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., Datta, Mitchell, Ramanathan, Journal of Cryptology 2008

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]

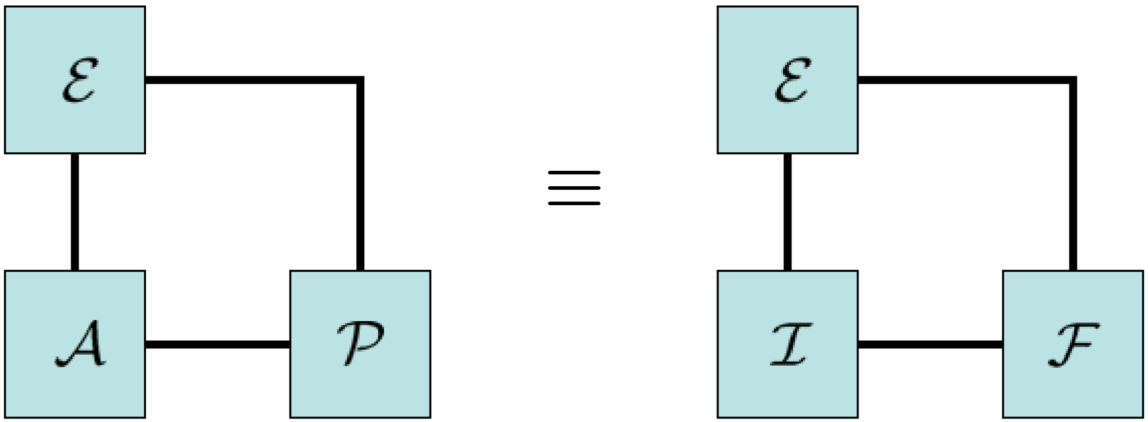
- Task PIOA
  [Canetti, Cheung, Kaynar, Liskov, Lynch, Peireira, Segala, 2006]

- IITM model
  [K., 2006]

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

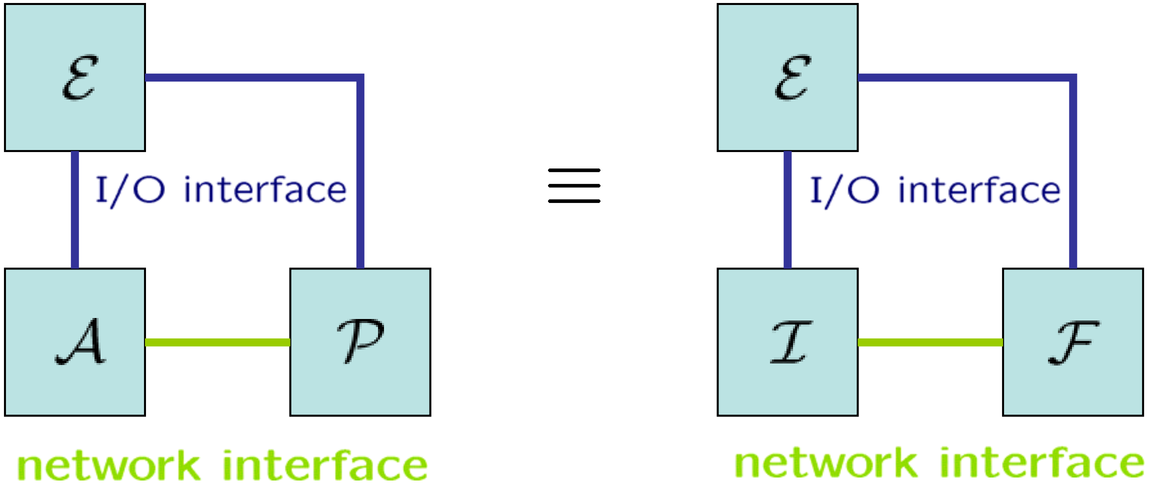
Universal Composability

$\mathcal{P}$ and $\mathcal{F}$ are UC if $\forall\, \mathcal{A}\ \exists\, \mathcal{I}\ \forall\, \mathcal{E}$:



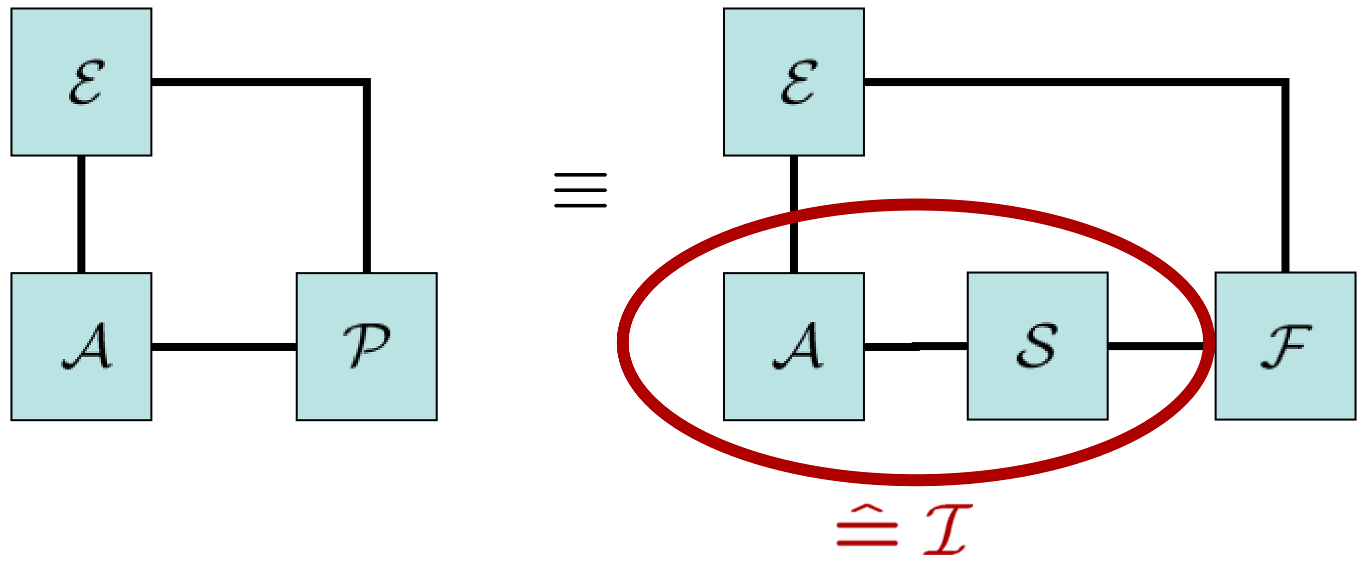
Universal Composability

$\mathcal{P}$ and $\mathcal{F}$ are UC if $\forall\, \mathcal{A}\ \exists\, \mathcal{I}\ \forall\, \mathcal{E}$:



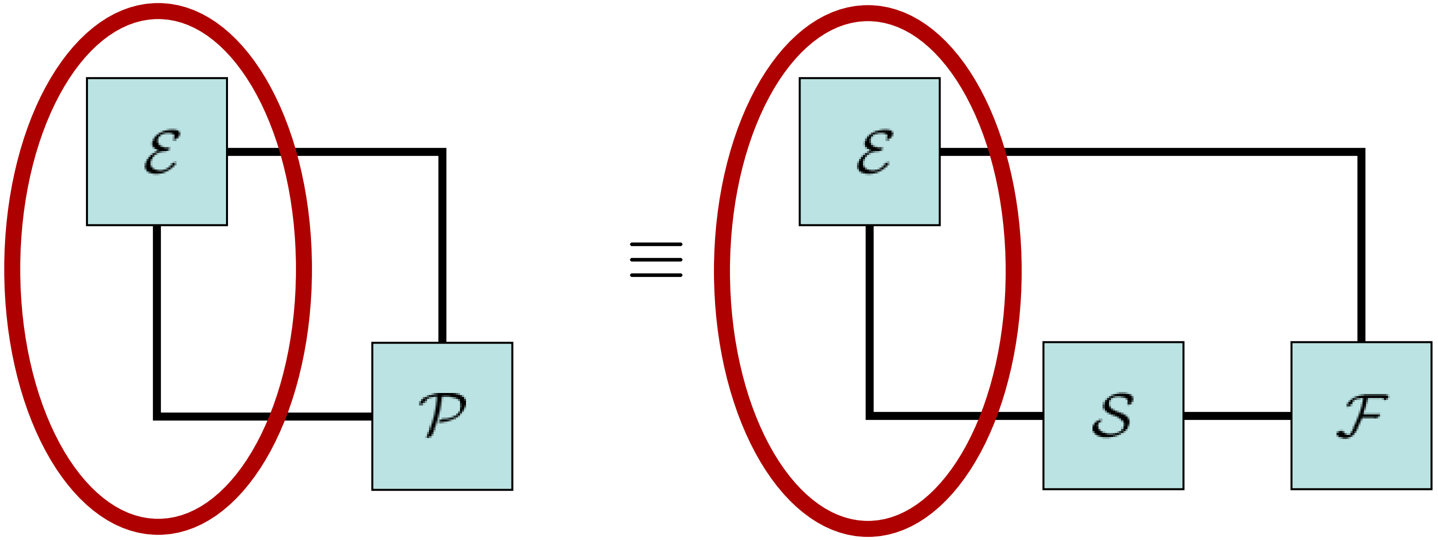
(Strong) Black-box Simulatability

$\mathcal{P}$ and $\mathcal{F}$ are SBB if $\exists\, \mathcal{S}\ \forall\, \mathcal{A}\ \forall\, \mathcal{E}$:



Strong Simulatability

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



$\widehat{=}\ \mathcal{E}\,||\,\mathcal{A}$ in UC and BB

Subtelties in Simulation-Based Models

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

... confusing at first sight

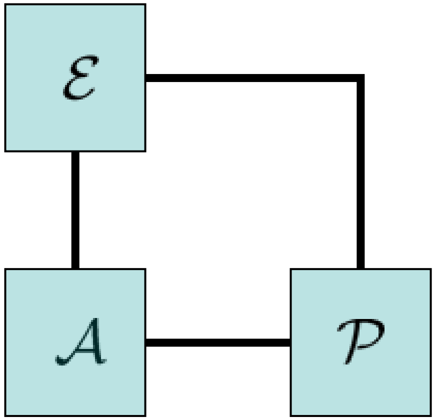
Let's look at two issues more closely

1. Master Process

2. Runtime of ITMs

Master Process

$\boxed{\text{Master process:}}$ 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

$SS \equiv SBB$

[PW 2001,BPW 2004]

Master: no restrictions

All of the following notions are equivalent:

SBB	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{E}$
Pfitzmann, Waidner 2001	X	X	
Backes, Pfitzmann, Waidner 2004	X	X	X
	X		X
		X	X
			X
	X		

SS	$\mathcal{S}$	$\mathcal{E}$
	X	X
		X

Relationships Between Security Notions

SS $\equiv$ SBB

[PW 2001,

Master: no

All of the following notions are equivalent:

UC	$\mathcal{A}$	$\mathcal{I}$	$\mathcal{E}$
Canetti 2001			X
Backes, Pfitzmann, Waidner 2004	X	X	X

WBB	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{E}$
	X	X	X
	X		X
		X	X
			X

UC $\equiv$ WBB

[C 2001, BPW 2004]

Master: environment + other entities

Relationships Between Security Notions

$SS \equiv SBB$

[PW 2001,BPW 2004]

Master: no restrictions

WBB	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{E}$
	X		

$UC \equiv WBB$

[C 2001, BPW 2004]

Master: environment + other entities

WBB

Master: only adversary

Relationships Between Security Notions

$\mathsf{SS} \equiv \mathsf{SBB}$

[PW 2001,BPW 2004]

Master: no restrictions

$\mathsf{UC} \equiv \mathsf{WBB}$

[PW 2001]

Master: not environment

All of the following notions are equivalent:

UC	$\mathcal{A}$	$\mathcal{I}$	$\mathcal{E}$
Pfitzmann, Waidner 2001	X	X	

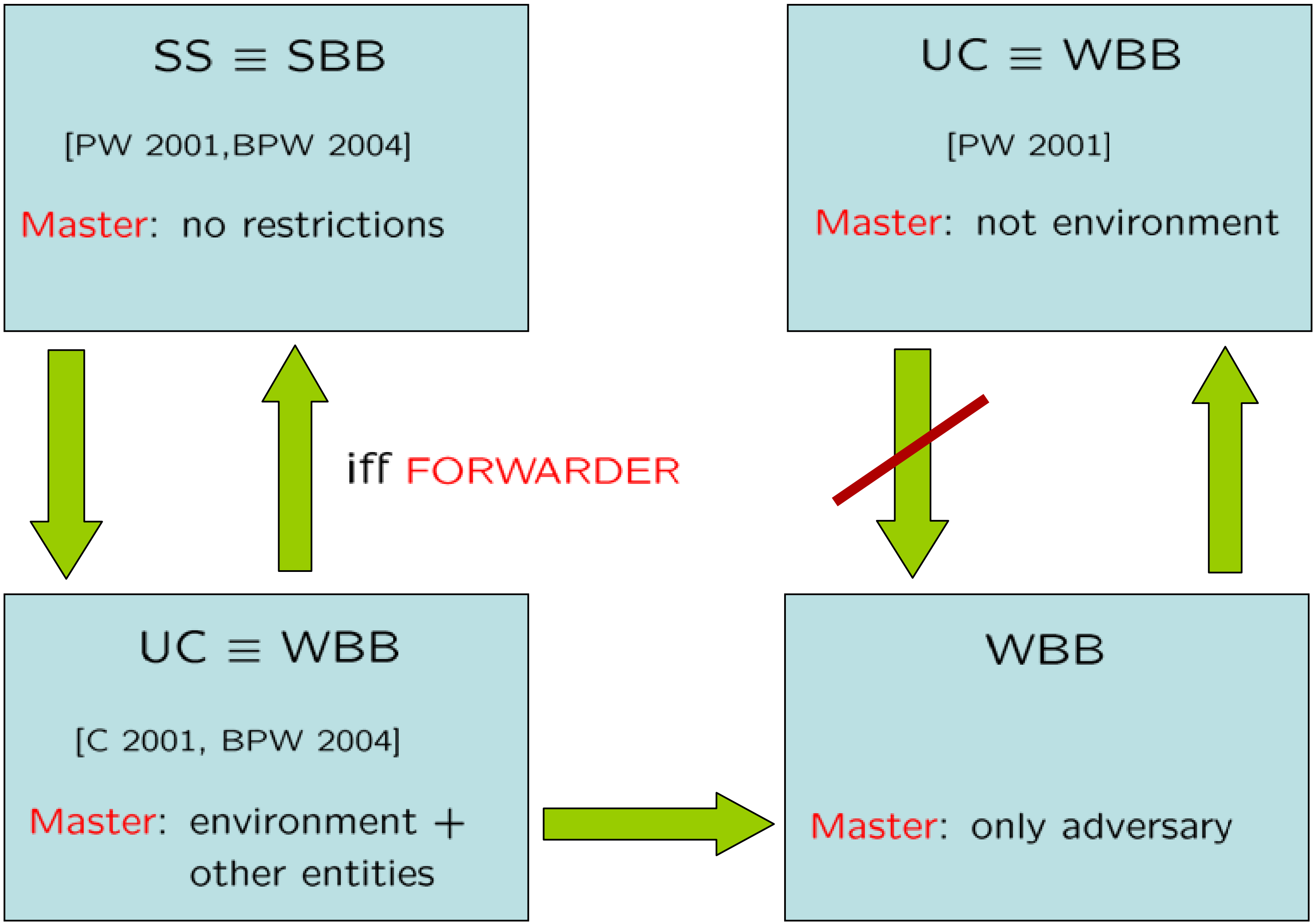
WBB	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{E}$
	X	X	

[C 2001, BPW 2004]

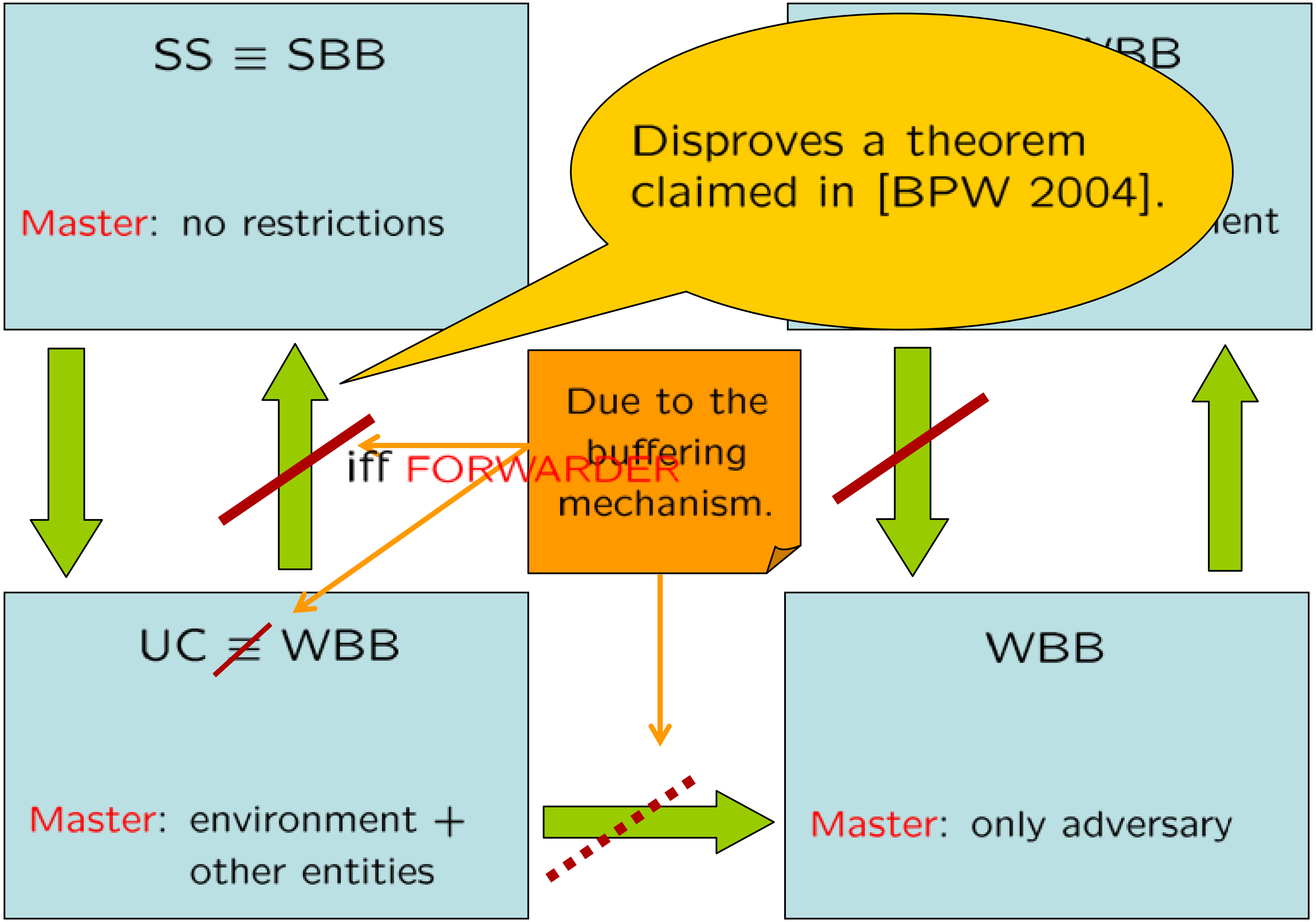
Master: environment + other entities

Master: only adversary

Relationships Between Security Notions



Relationships in BPW Model



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.

⇓

Systems run in polynomial time.

  - Probab Polynomial Time Process Calculi.
  - All t ove 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:

$$\mathcal{P} \,||\, \mathcal{Q} \;\equiv\; \mathcal{P} \,||\, \mathcal{D} \,||\, \mathcal{Q}.$$

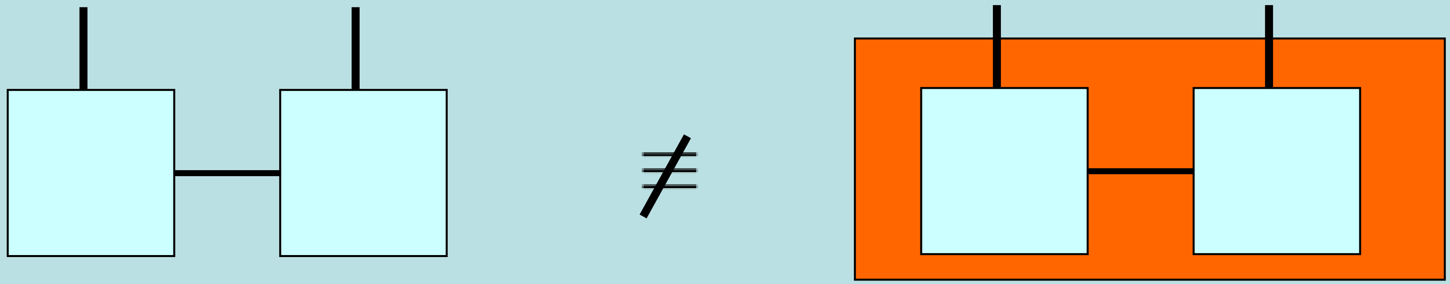
[Datta, Küsters, Mitchell, Ramanathan 2005]

Unintuitive a… 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):



But simulation possible with length functions or guards.

- Almo[illegible]entical 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]).

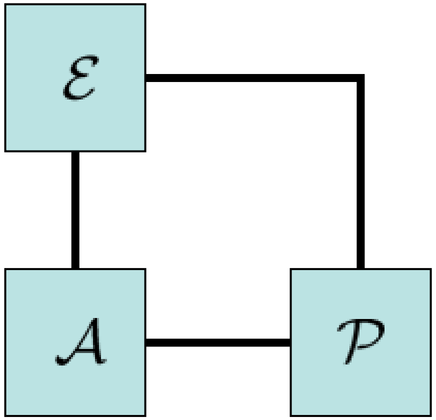
Overview

- 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 [K., 2006]

- 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 [K., 2006]

- 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

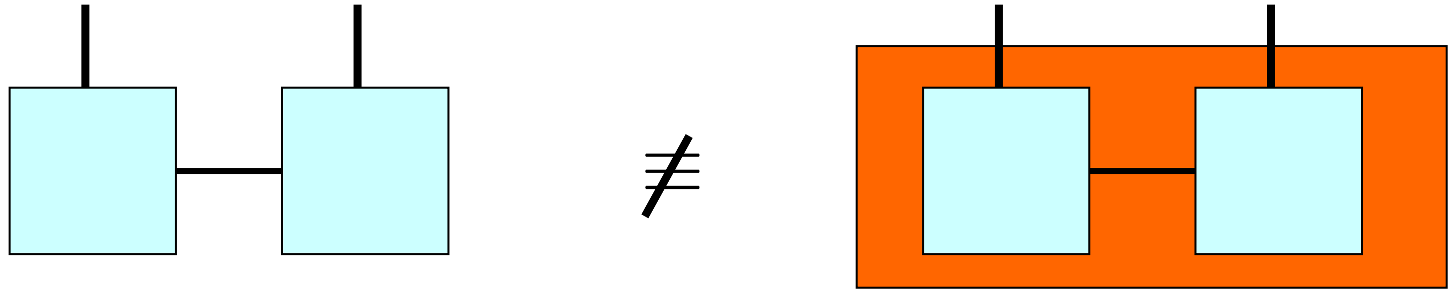
  (no specific addressing mechanism, e.g., based on SIDs/PIDs, is fixed)

- Runtime may depend on length of input
- Can be activated an unbounded number of times
- Can perform ppt computation in *every* activation

  $\Rightarrow$ no exhaustion

yields more useful functionalities and more natural properties

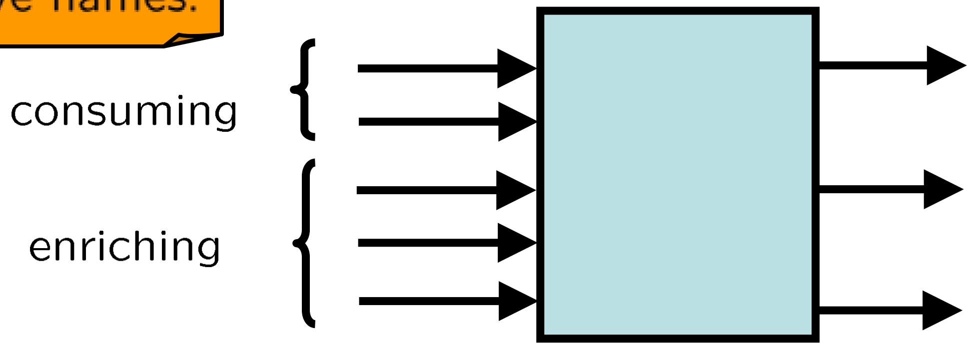
If machines could be forced to stop (e.g., UC model):



two machines $\neq$ single machine

Inexhaustible ITMs (IITMs)

Tapes have names.



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

$\Rightarrow$ ITM can read every message and can scan entire configuration in every activation.

$\Rightarrow$ 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

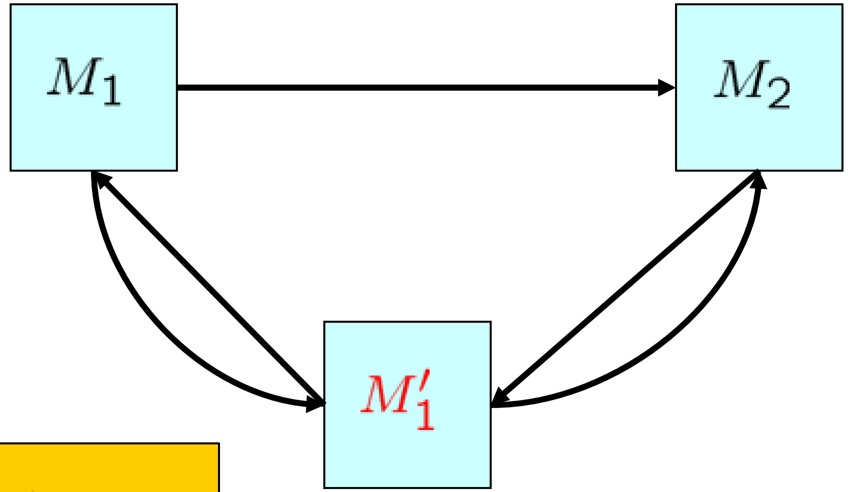
Systems of IITMs

This is not process calculus

$$\mathcal{S} = M_1 \,\|\, \cdots \,\|\, M_n \,\|\, !M'_1 \,\|\, \cdots \,\|\, !M'_m$$

single IITM

unbounded number of copies generated dynamically



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 $\mathcal{D}$:

$$\boxed{\mathcal{P} \,||\, \mathcal{Q} \quad \equiv \quad \mathcal{P} \,||\, \mathcal{D} \,||\, \mathcal{Q}}$$

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

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

$$\boxed{\mathcal{Q}_1 \,||\, \mathcal{Q}_2 \;\equiv\; \mathcal{Q}_1 \,||\, M}$$

(Needed, e.g., in joint state theorem)

Copies of IITMs and the Generic Addressing Mechanism

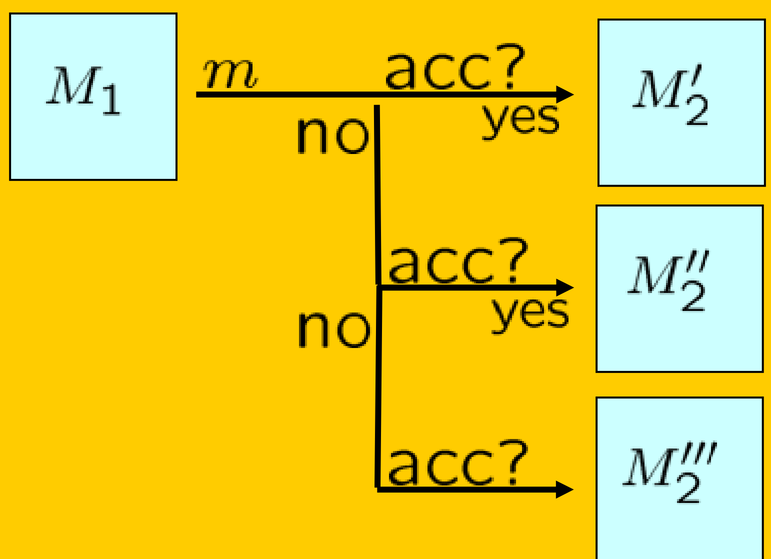
IITMs run in one of two modes:

Check address
Compute

$\Bigg\}$ generic addressing mechanism

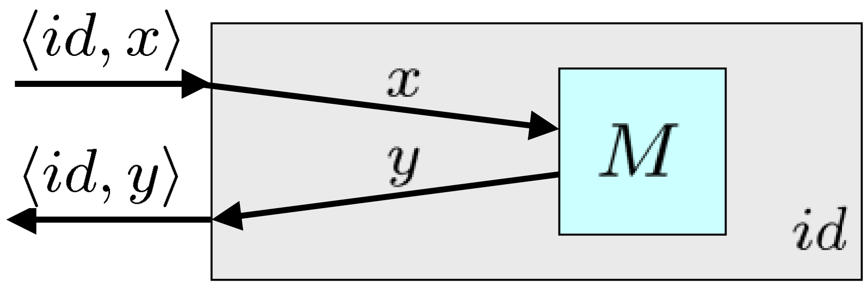
Example: $\mathcal{S} = M_1 \,\|\, !M_2$

$M_1 \xrightarrow{c} M_2$



Copies of IITMs and the Generic Addressing Mechanism

ID version of M: (allows to address multiple copies of M)



$= \underline{M}$

ID can be SID (session version) or PID (party version) of M

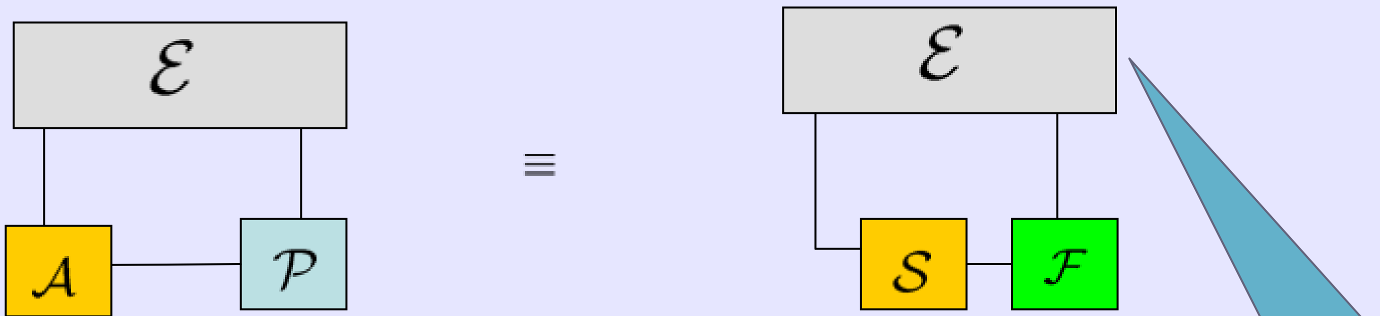
The IITM Model

- General computational model
  - Inexhaustible Interactive Turing Machines (IITMs)
  - System of IITMs
- Simulation-based security
  - Security notions
  - Composition theorems

Security Notions

Definition (Strong Simulatability)

$\mathcal{P} \leq \mathcal{F}$ iff $\exists$ Sim. $\mathcal{S}$ $\forall$ Env. $\mathcal{E}$: $\mathcal{E} \,||\, \mathcal{P} \equiv \mathcal{E} \,||\, \mathcal{S} \,||\, \mathcal{F}$



master system

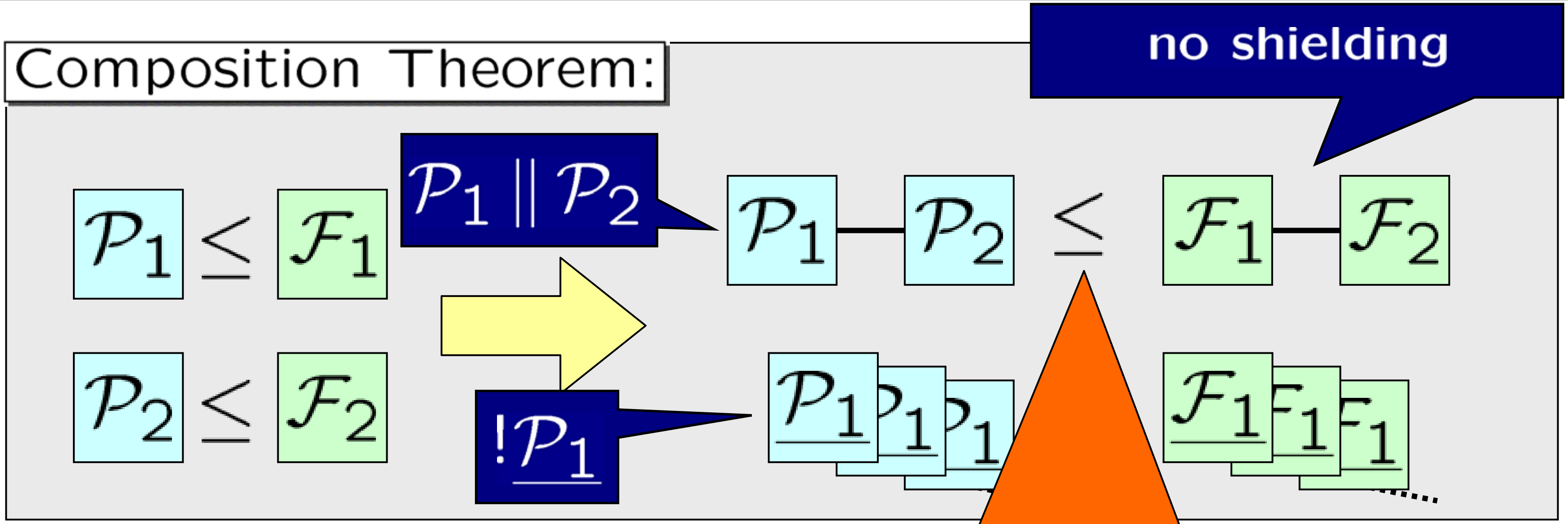
- Similarly for UC and black-box simulatability

- No unnecessary details need to be fixed in the IITM model

  * Corruption
  * Addressing

  Part of the description of protocols ($\Rightarrow$ flexible and expressive)

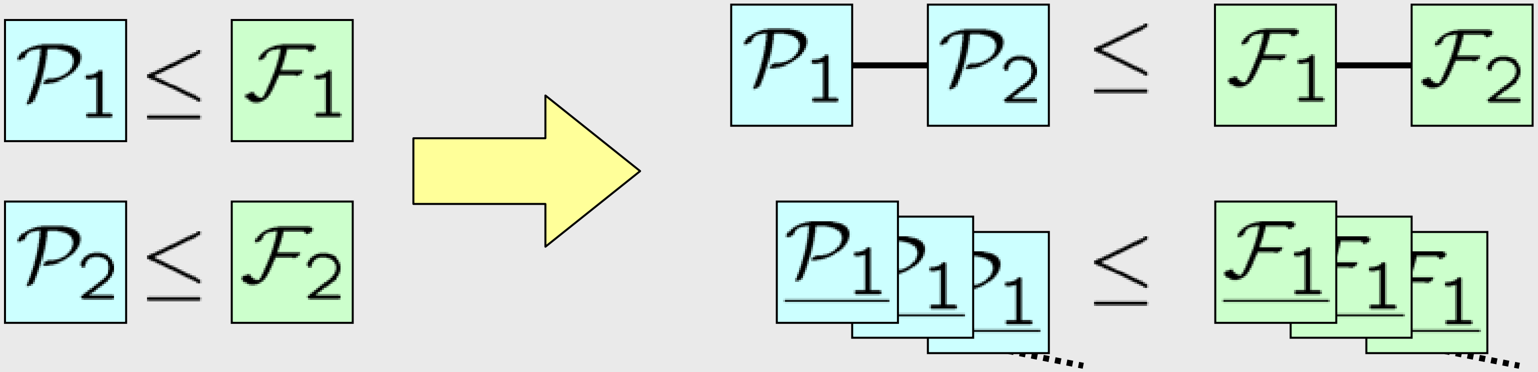
Composition Theorem in IITM Model



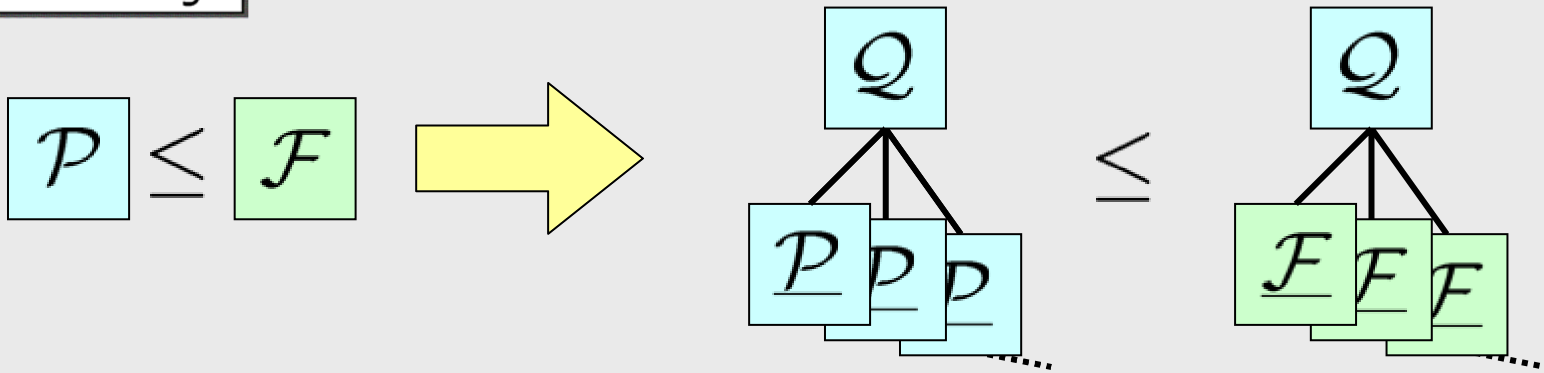
$$
\begin{aligned}
\mathcal{E} \,\|\, \mathcal{P}_2 \,\|\, \mathcal{P}_1 &\equiv [\mathcal{E} \,\|\, \mathcal{P}_2]_{P_1} \,\|\, \mathcal{P}_1 \\
&\equiv [\mathcal{E} \,\|\, \mathcal{P}_2]_{P_1} \,\|\, \mathcal{S}_1 \,\|\, \mathcal{F}_1 \\
&\equiv \mathcal{E} \,\|\, \mathcal{P}_2 \,\|\, \mathcal{S}_1 \,\|\, \mathcal{F}_1 \\
&\equiv \mathcal{E} \,\|\, \mathcal{S}_1 \,\|\, \mathcal{F}_1 \,\|\, \mathcal{P}_2 \\
&\equiv [\mathcal{E} \,\|\, \mathcal{S}_1 \,\|\, \mathcal{F}_1]_{P_2} \,\|\, \mathcal{P}_2 \\
&\equiv [\mathcal{E} \,\|\, \mathcal{S}_1 \,\|\, \mathcal{F}_1]_{P_2} \,\|\, \mathcal{S}_2 \,\|\, \mathcal{F}_2 \\
&\equiv \mathcal{E} \,\|\, \mathcal{S}_1 \,\|\, \mathcal{F}_1 \,\|\, \mathcal{S}_2 \,\|\, \mathcal{F}_2 \\
&\equiv \mathcal{E} \,\|\, \mathcal{S}_1 \,\|\, \mathcal{S}_2 \,\|\, \mathcal{F}_1 \,\|\, \mathcal{F}_2
\end{aligned}
$$

Composition Theorem in IITM Model

Composition Theorem:



Corollary:



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

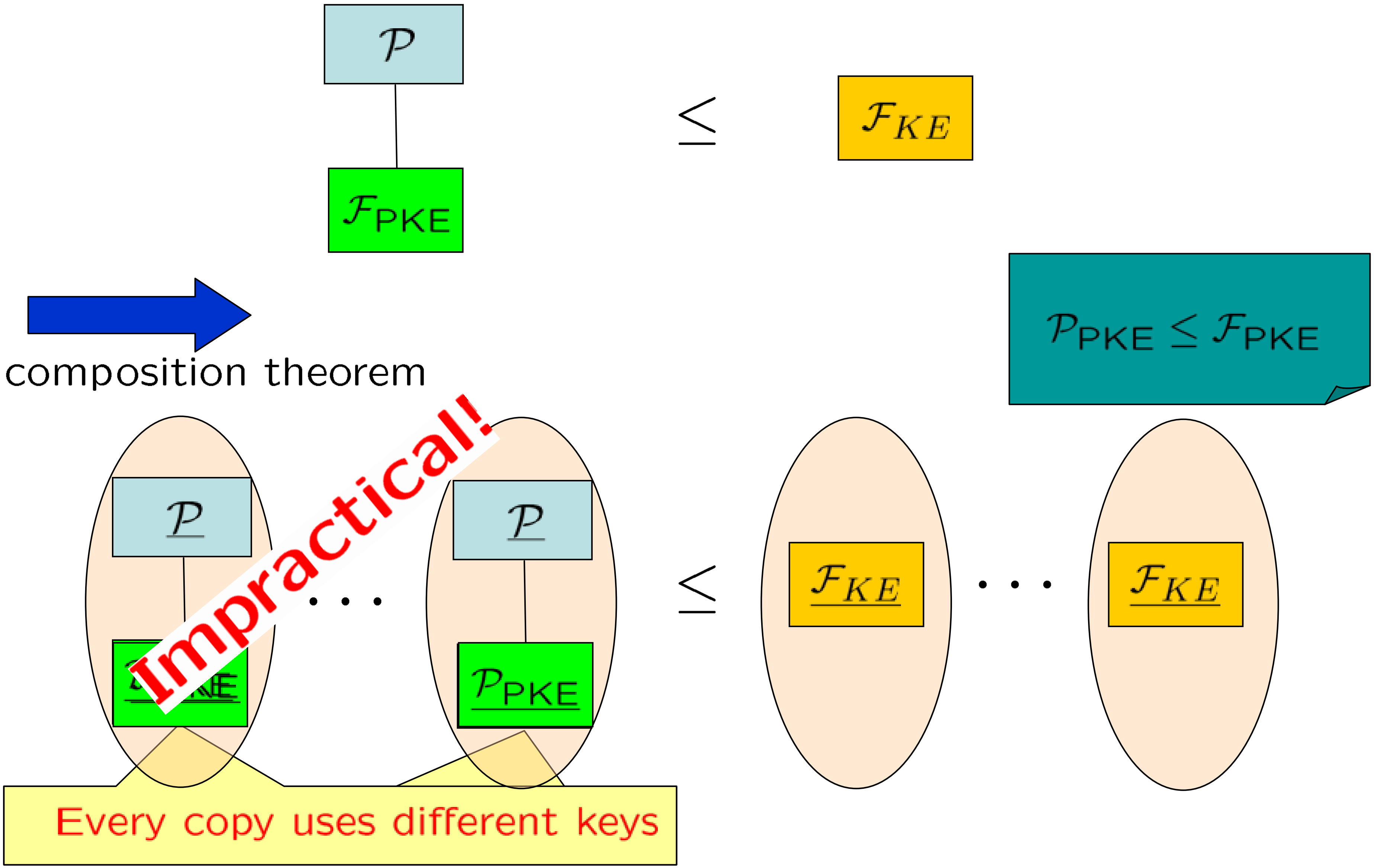
Models for Simulation-Based Security

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

Joint State Theorem

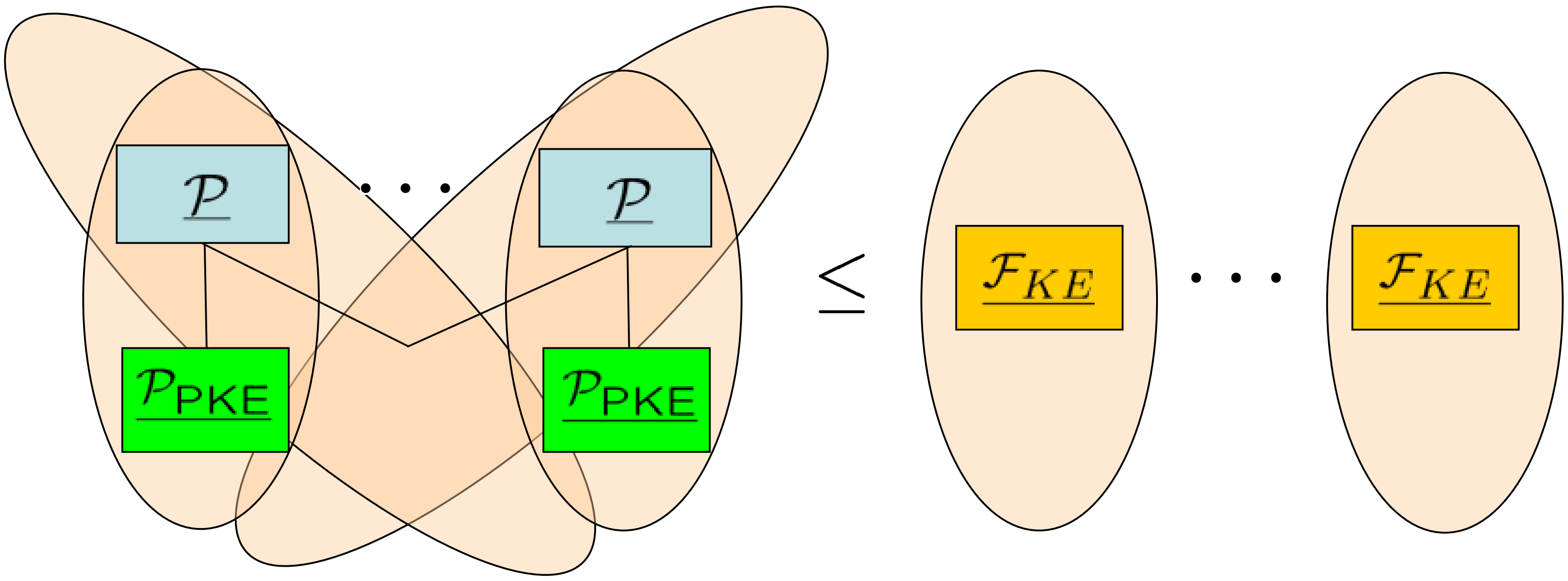
Comparison:
[Datta, Küsters, Mitchell, Ramanathan 2005/2008]

Motivating Joint State



Motivating Joint State

We rather want:



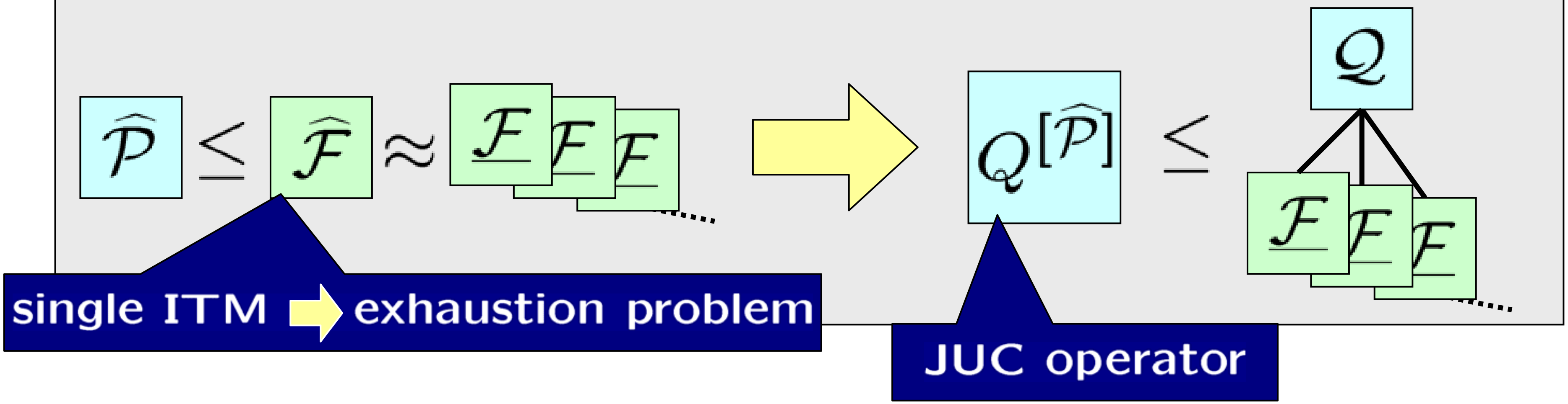
Joint State

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]



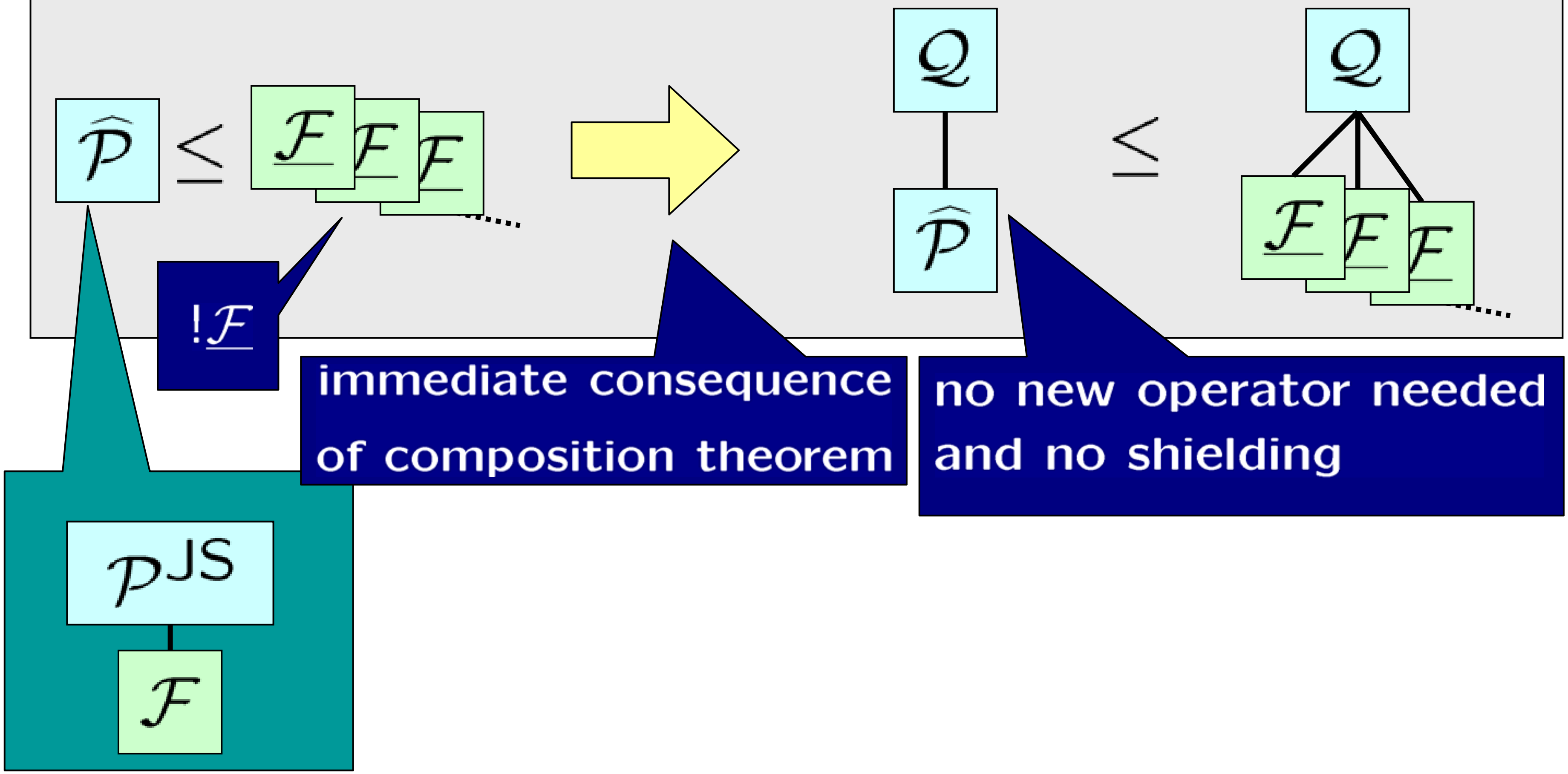
Conceptually a good idea,

but technically the theorem is flawed

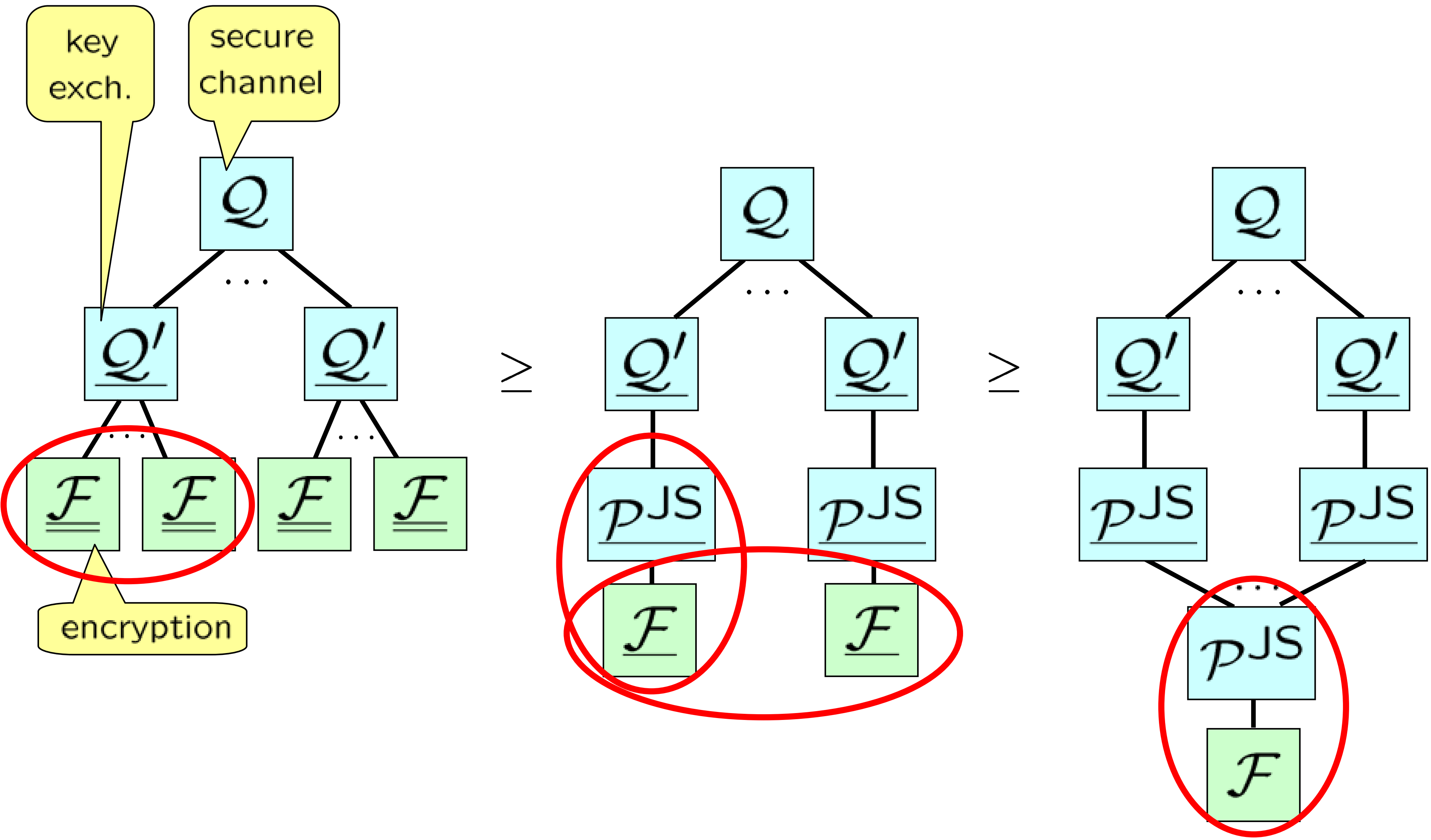
General Joint State Theorem – IITM Model

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

$\widehat{\mathcal{P}} \leq !\underline{\mathcal{F}}$ $\Rightarrow$ $\mathcal{Q} \,||\, \widehat{\mathcal{P}} \leq \mathcal{Q} \,||\, !\underline{\mathcal{F}}$



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: $\mathcal{F}_{\mathrm{PKE}}(l)$

$\mathcal{F}_{\mathrm{PKE}}$: used by one decryptor and arbitrary many encryptors.

$l\colon \{0,1\}^* \to \{0,1\}^*$ models **leakage**, e.g., $l_1 : m \mapsto 1^{|m|}$

Literature:

Now:

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

ge
for JS

En

if

else

$c := e(k'_e, m)$

$\mathcal{F}_{\mathrm{PKE}}$

ed

error

store (m, c)

else

$c := e(k'_e, m)$

$\mathcal{F}_{\mathrm{P}}$ (l)

important for JS
missing in other formulations

e and d are provided by the simulat

Realizing $\mathcal{F}_{\mathsf{PKE}}$ by CCA-Secure Encryption Schemes

Theorem: [K. and Tuengerthal 2008]

Σ is IND-CCA $\Longleftrightarrow$ $\boxed{\mathcal{P}_{\mathsf{PKE}}(\Sigma)} \leq \boxed{\mathcal{F}_{\mathsf{PKE}}(l)}$

realization corresponding to Σ

Joint State Realization for PKE

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

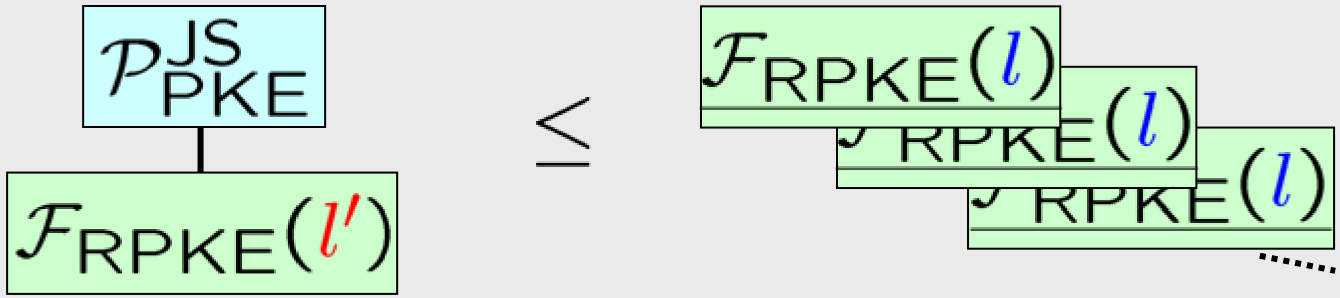
$$\frac{\mathcal{P}^{\text{JS}}_{\text{PKE}}}{\mathcal{F}_{\text{PKE}}(l')} \;\leq\; \underline{\mathcal{F}_{\text{PKE}}(l)}\;\; \underline{\mathcal{F}_{\text{PKE}}(l)}\;\; \underline{\mathcal{F}_{\text{PKE}}(l)}\;\cdots$$

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

- Basic idea of $\mathcal{P}^{\text{JS}}_{\text{PKE}}$,
  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:

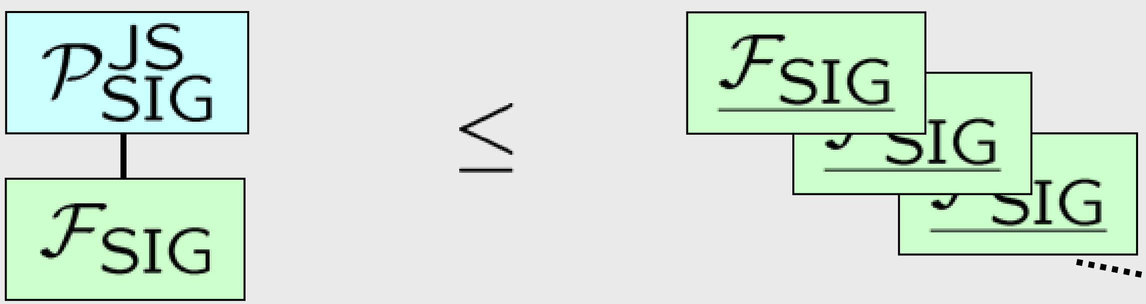


where $l'(\langle \mathsf{sid}, m\rangle) = \langle \mathsf{sid}, l(m)\rangle$ (leakage of SID)

plus realization for $\mathcal{F}_{\mathsf{RPKE}}$ for IND-RCCA secure Σ

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

Joint State Theorem for Non-Interactive Digital Signatures:



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
- ...

Except for a Dolev-Yao style functionality [Backes, Pfitzmann '04], see related work

But not for symmetric key encryption!

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)

More complicated than for **PKE** where **private keys stay in functionality**

- **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

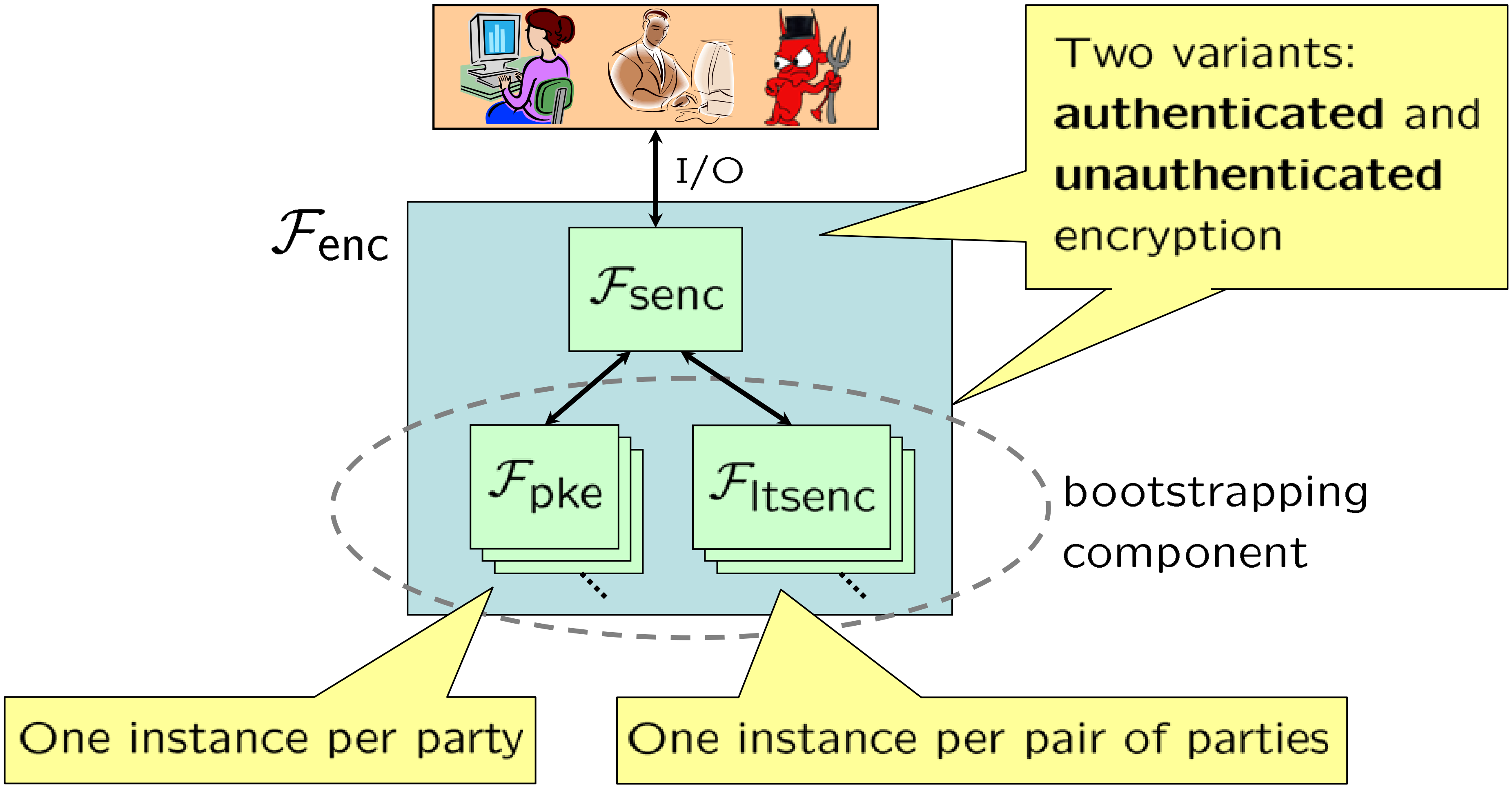
2. Our symmetric key encryption functionality

3. Applications of the functionality

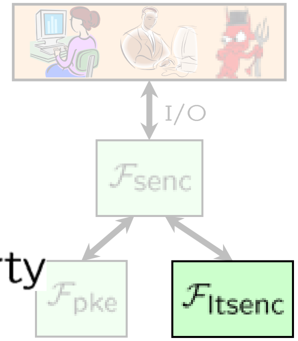
   - Proving a protocol secure
   - Simplifying game-based proofs
   - Computational soundness for UC realization of key exchange

4. Related work

Our Symmetric Key Encryption Functionality



Our Symmetric Key Encryption Functionality



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

- User commands (I/O):
  - Key exchange: Ask for a key to be exchanged with other party
  - Encrypt m: If corrupt: Return $c = enc(m)$
    Else: Return $c = enc(L(m))$ and record (m, c)

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

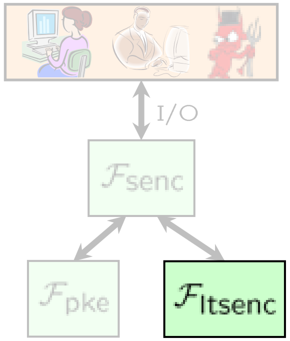
  - Decrypt c: If corrupt: Return $m = dec(c)$
    Else: Return $m = \begin{cases} m & \text{if recorded } (m,c) \\ dec(c) & \text{if variant } \mathcal{F}^{\text{unauth}}_{\text{ltsenc}} \text{ and } c \text{ not recorded} \\ \perp & \text{otherwise} \end{cases}$

No assumptions
Hence, abstract from algorithms

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

Our Symmetric Key Encryption Functionality

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



- Σ symmetric encryption scheme
- Induces the obvious realization $\mathcal{P}(\Sigma)$

(no extra randomness or tagging)

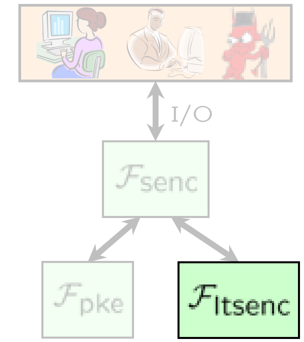
Theorem:

Σ IND-CCA secure $\Leftrightarrow$ $\boxed{\mathcal{P}(\Sigma)} \leq \boxed{\mathcal{F}^{\text{unauth}}_{\text{ltsenc}}}$

Σ authenticated encryption scheme (IND-CPA + INT-CTXT secure) $\Leftrightarrow$ $\boxed{\mathcal{P}(\Sigma)} \leq \boxed{\mathcal{F}^{\text{auth}}_{\text{ltsenc}}}$

Our Symmetric Key Encryption Functionality

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



Multi-session case: Composition theorem yields

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

Impractical:
Different key for each session

Joint State Theorem:

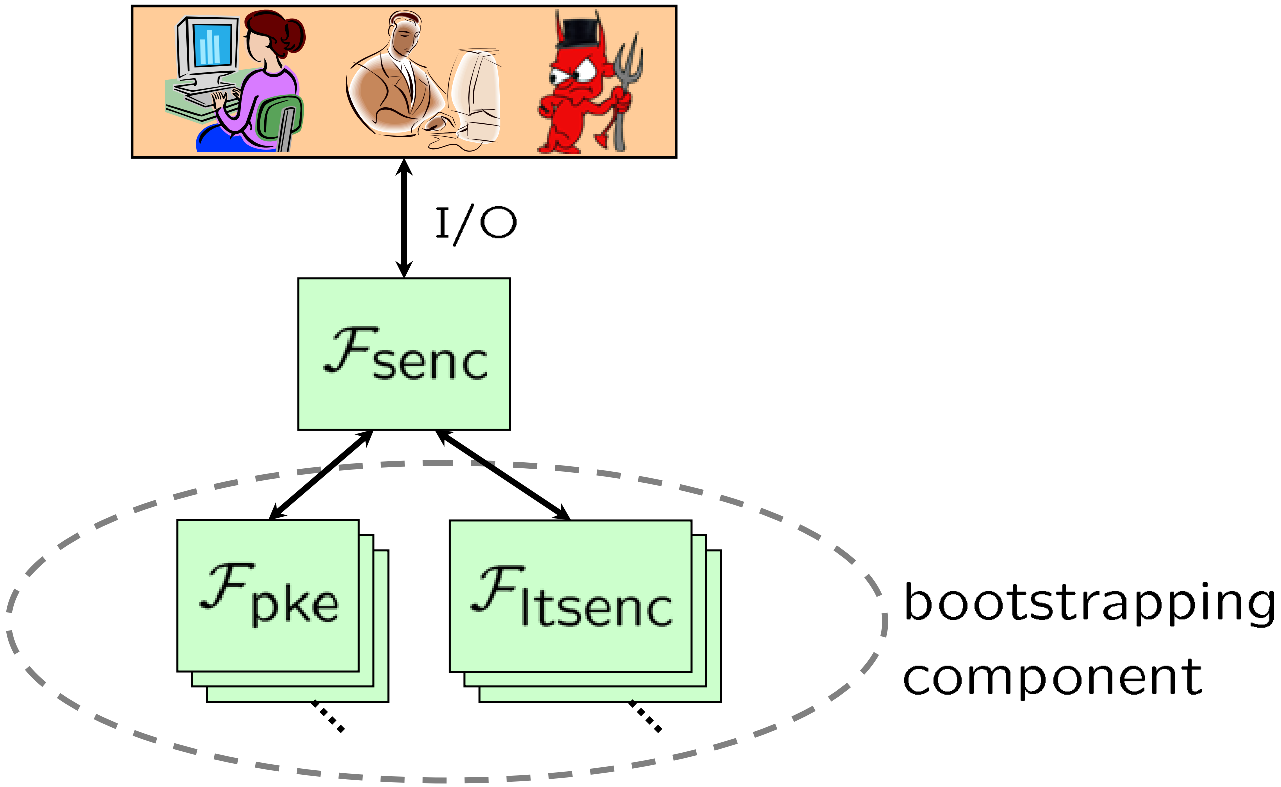
Encrypts $\langle \text{sid}, m \rangle$ instead of m

$$\begin{array}{c} \mathcal{P}^{\text{js}}_{\text{ltsenc}} \\ \updownarrow \\ \mathcal{F}_{\text{ltsenc}} \end{array} \leq \mathcal{F}_{\text{ltsenc}}$$

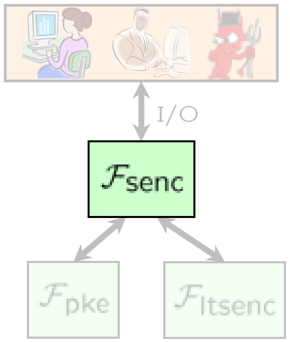
One instances
per pair of parties

Holds for both variants **unauthenticated** and **authenticated** encryption

Our Symmetric Key Encryption Functionality



Our Symmetric Key Encryption Functionality



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

Pointer management:

- For each party, mapping from pointers ($\mathbb{N}$) to keys (bit strings)

$$p \mapsto k_p$$

- Plaintexts m are arbitrary bit strings, may contain "(KeyPtr, p)"

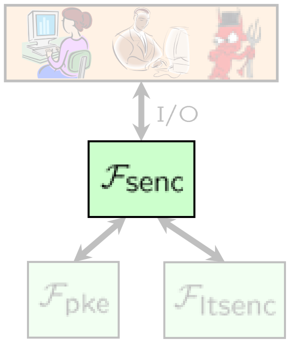
  Replace "(KeyPtr, p)" by "(Key, k_p)" before encryption

  Replace "(Key, k_p)" by "(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 $\mathcal{F}_{\text{senc}}$:



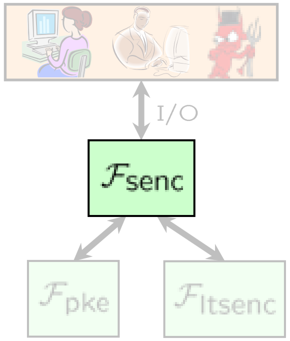
- User commands (I/O):
  - Generate Key: Adversary provides key k (bit string)
    New pointer p is returned to user
  - Encrypt/Decrypt: Encrypt $L(m)$ if key is "unknown",
    else encrypt m

> 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: $enc(\cdot,\cdot), dec(\cdot,\cdot)$
  - Corrupt: Static corruption (keys corruptible upon generation)

Our Symmetric Key Encryption Functionality

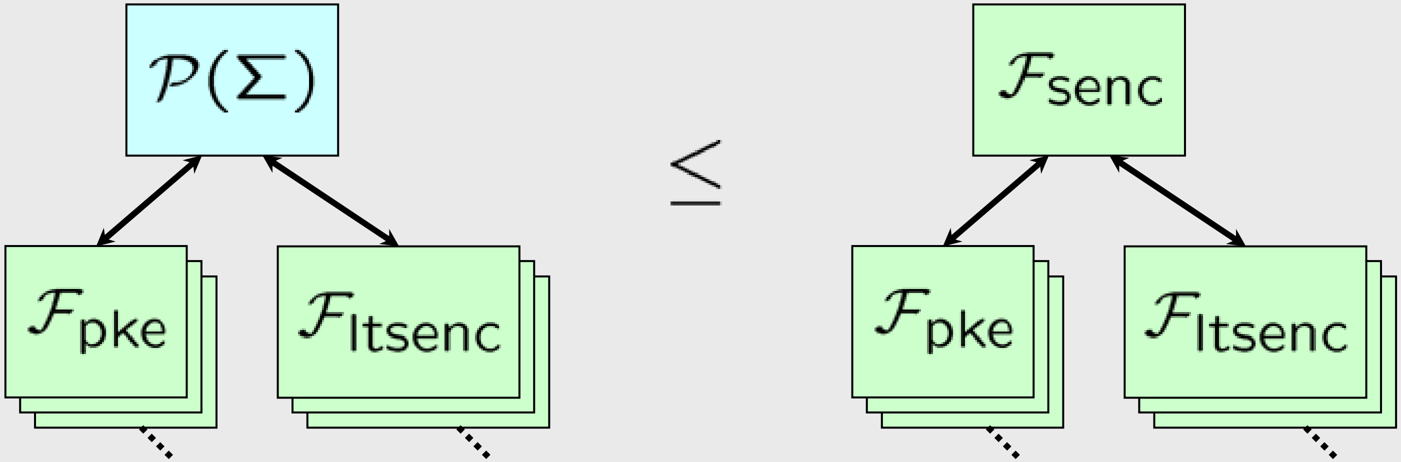
Realizing $\mathcal{F}_{\text{senc}}$:



- Σ symmetric encryption scheme
- Induces the obvious realization $\mathcal{P}(\Sigma)$

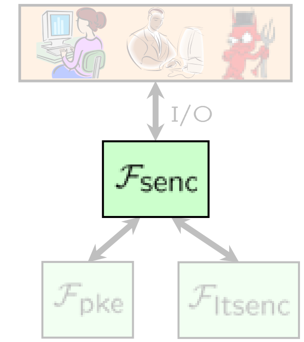
(no extra randomness or tagging)

We would like to prove:



Impossible because of **key cycle** and **commitment** problems

Our Symmetric Key Encryption Functionality



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

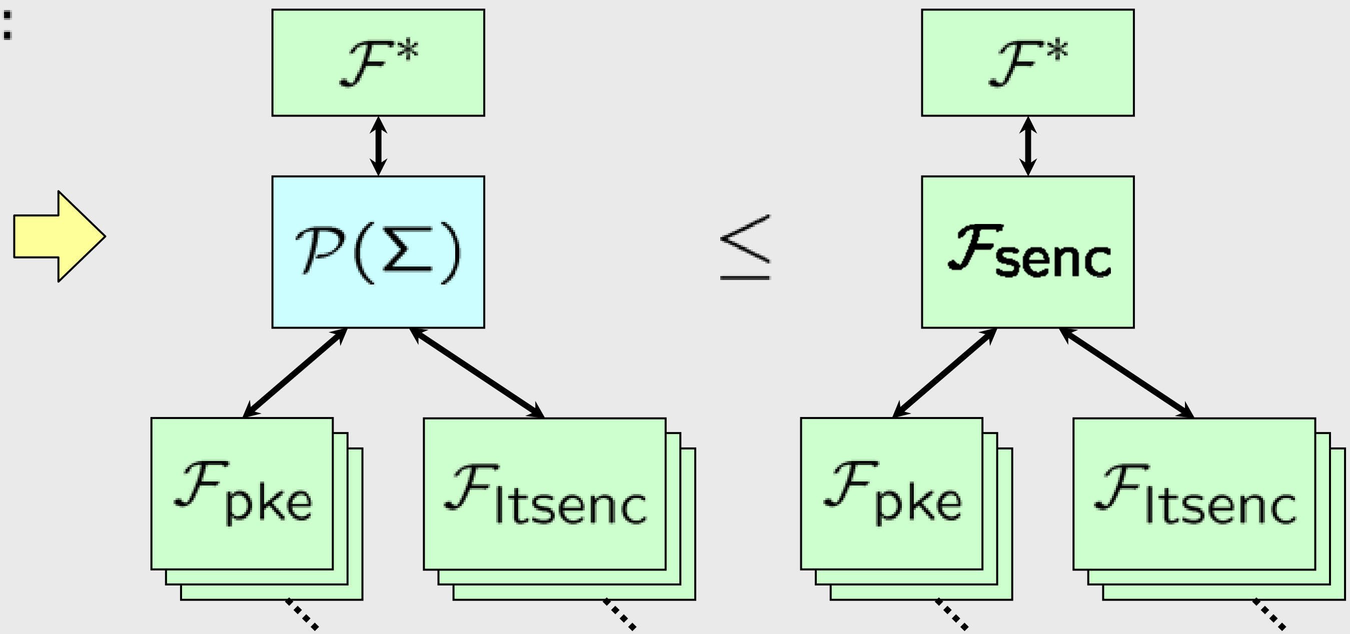
- Restricting the environment:
  1. Used order respecting: [Backes, Pfitzmann '04]

     Keys ordered by first use

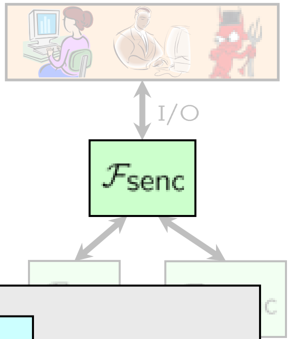
     Keys may only encrypt later keys
  2. Non-committing: An unknown used key must not become known

Main Theorem:

Σ IND-CCA (resp., authenticated encryption)



Our Symmetric Key Encryption Functionality



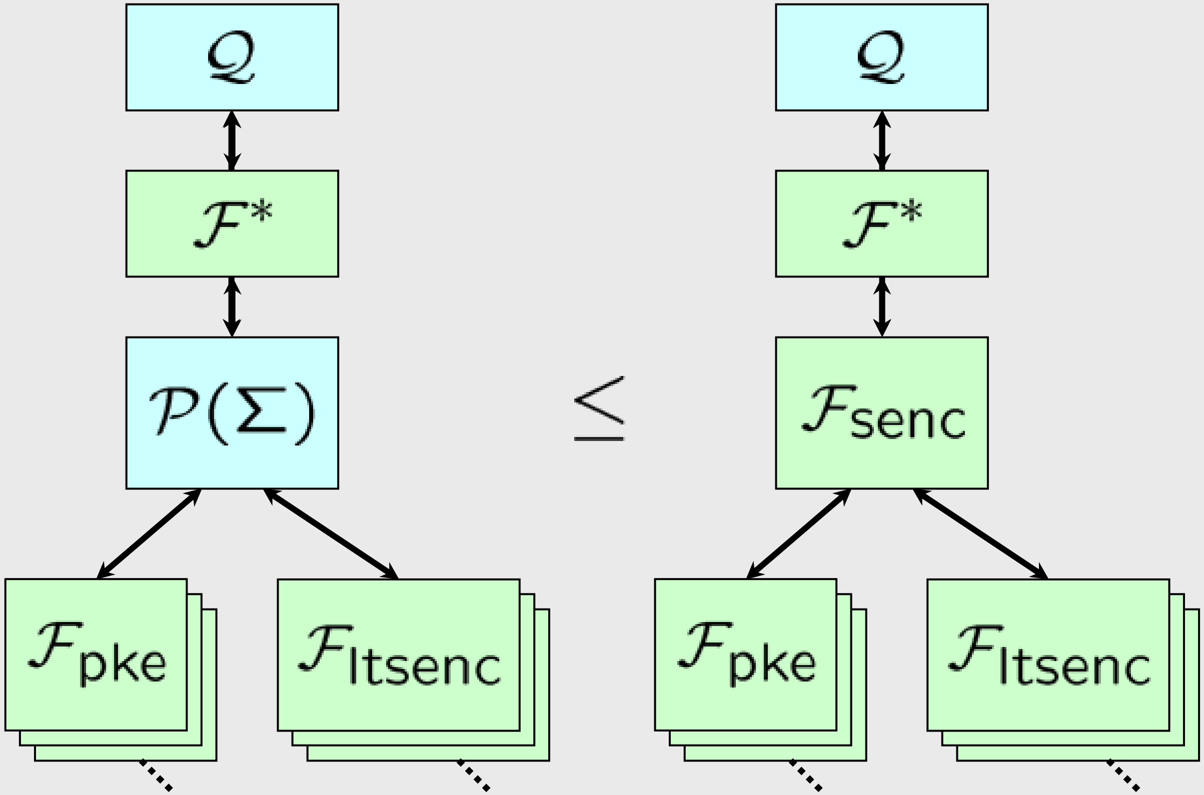
Realizing $\mathcal{F}_{\text{senc}}$:

Typically, a protocol $\mathcal{Q}$ enforces these restrictions:

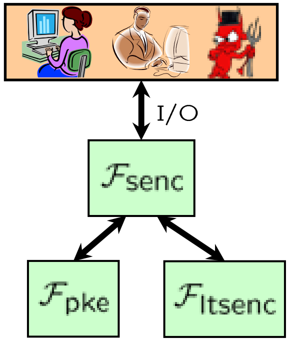
(e.g. Kerberos)

Corollary:

Σ IND-CCA (resp., authenticated encryption)



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 $\mathcal{F}_{\text{ltsenc}}$ 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 Amanded NSSK protocol:

1. $B \to A\colon \{A, k_B\}_{k_{BS}}$
2. $A \to S\colon A, B, n_A, \{A, k_B\}_{k_{BS}}$
3. $S \to A\colon \{n_A, B, k_{AB}, \{k_{AB}, A\}_{k_B}\}_{k_{AS}}$
4. $A \to B\colon \{k_{AB}, A\}_{k_B}$

k_B – short-term key

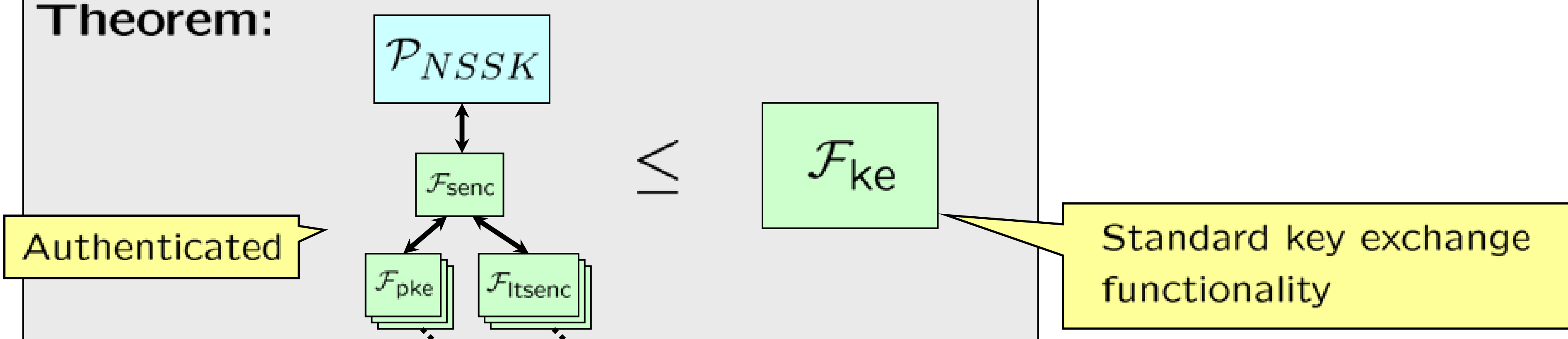
k_{AS}, k_{BS} – long-term keys

k_{AB} – session key

n_A – nonce

- Formulate as protocol system $\mathcal{P}_{NSSK}$ which uses $\mathcal{F}_{\text{senc}}$

Theorem:



Applications: Proving a Protocol Secure

Proof of theorem:

- By joint state and compostion theorem
  we only need to consider a **single instance** between A, B and S
- If somebody is corrupt, simulator can corrupt $\mathcal{F}_{\text{ke}}$ $\Rightarrow$ done
- If all are uncorrupted:

  1. $B \rightarrow A\colon \{A, k_B\}_{k_{BS}}$
  2. $A \rightarrow S\colon A, B, n_A, \{A, k_B\}_{k_{BS}}$
  3. $S \rightarrow A\colon \{n_A, B, k_{AB}, \{k_{AB}, A\}_{k_B}\}_{k_{AS}}$
  4. $A \rightarrow B\colon \{k_{AB}, A\}_{k_B}$

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

1. S uncorrupt $\Rightarrow$ $k_{AB} = k'_{AB}$
2. All keys are unknown
   $\Rightarrow$ k_{AB} encrypted ideally
   $\Rightarrow$ indistuinguishable from random

Applications: Simplifying Game-based Proofs

Secretive protocols:

[Roy, Datta, Derek, Mitchell '07]

- A protocol $\mathcal{P}$ is **secretive** w.r.t. a key k if k is only sent "properly" encrypted

Typically:
$\mathcal{P}$ KE protocol
k the session key

- $\mathcal{P}$ secretive w.r.t. k

  ⇒ **key usability** for k

  ⇒ **key indistinguishability** for k if k is not used in protocol

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

- Definition: $\mathcal{P}$ 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

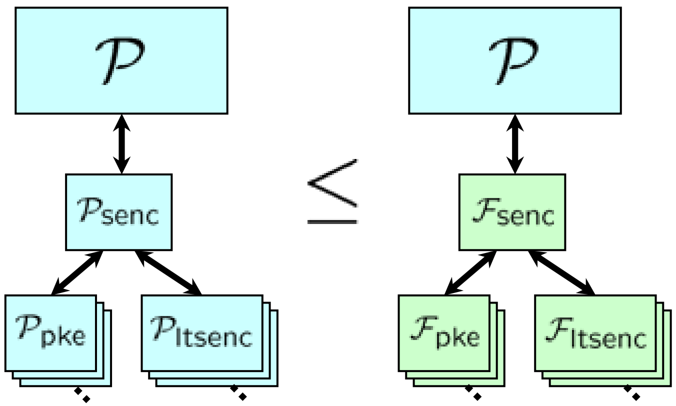
Theorem:

$\mathcal{P}$ is **secretive** w.r.t. k

⇒ **key usability** for k

⇒ **key indistinguishability** for k if k is not used in protocol

Proof: (for key indistinguishability)



In the **ideal** world:

k is never used

k is only encrypted ideally

⇒ k is indistinguishable from random

⇒ k is indistinguishable from random in the **real** world

Proof for key usability: Similarly simple

Using $\mathcal{F}_{\mathsf{senc}}$ 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)

easy to check automatically

- 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

information theoretic proof

observational equivalence

can be check automatically

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

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

then

authenticated encryption version

$\mathcal{P}$

$\mathcal{F}_{\text{enc}}$

$\leq$

$\mathcal{F}_{\text{ke}}$

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

$\mathcal{F}_{\text{pke}}$ $\mathcal{F}_{\text{ltsenc}}$

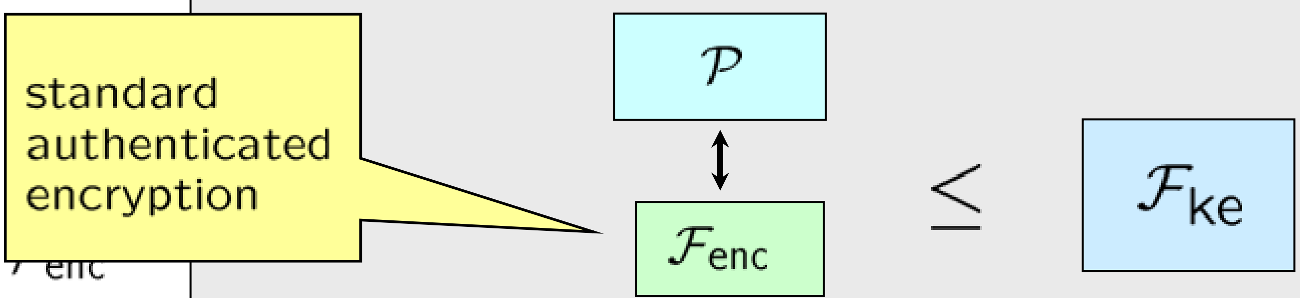
standard key exchange functionality

Using $\mathcal{F}_{\text{senc}}$ for Computational Soundness

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

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

then



Immediate consequence of composition theorem

Using $\mathcal{F}_{\mathsf{senc}}$ for Computational Soundness

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

$$!\,\underline{\mathcal{P} \,|\, \mathcal{P}_{\mathsf{enc}}} \;\leq\; !\,\underline{\mathcal{F}_{\mathsf{ke}}}$$

multi session version

Immediate consequence of composition theorem

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

Practical realization: replace multi session version of $\mathcal{F}_{\mathsf{ltsenc}}/\mathcal{P}_{\mathsf{ltsenc}}$ 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**

But:
- 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