Sequential Process Calculus and Machine Models for Simulation-based Security

Ralf Küsters

University of Kiel

Joint work with Anupam Datta, John Mitchell, and Ajith Ramanathan

Simulation-based Security

Basic idea:

1. Describe security requirement in terms of an ideal protocol/functionality $\mathcal{F}$.
2. A real protocol $\mathcal{P}$ is secure w.r.t. $\mathcal{F}$ (realizes $\mathcal{F}$) if everything that can happen to $\mathcal{P}$ can also happen to $\mathcal{F}$.
3. Goal: Security preserved under composition (composition theorem).

Simulation-based Security

Basic idea:

1. Describe security requirement in terms of an ideal protocol/functionality $\mathcal{F}$.
2. A real protocol $\mathcal{P}$ is secure w.r.t. $\mathcal{F}$ (realizes $\mathcal{F}$) if everything that can happen to $\mathcal{P}$ can also happen to $\mathcal{F}$.
3. Goal: Security preserved under composition (composition theorem).

But... Many different computational settings and security notions.

Canetti 2001 (PITM)

Computational model:

1. Computational entities:

   Probabilistic polynomial-time interacting turing machines (PITMs)

2. Communication model:

   In a real, ideal, and hybrid model specific ways of communication via tapes between an environment, a (real/ideal) adversary, and the (real/ideal) protocol are defined.

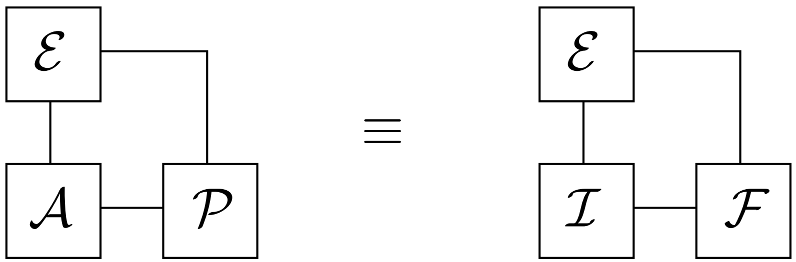
Canetti 2001 (PITM)

Computational model:

1. Computational entities:
   Probabilistic polynomial-time interacting turing machines (PITMs)
2. Communication model:
   In a real, ideal, and hybrid model specific ways of communication via tapes between an environment, a (real/ideal) adversary, and the (real/ideal) protocol are defined.

Security notion: Universal composability (UC).

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



Pfitzmann and Waidner 2001 (PIOA)

Computational model:

1. Computational entities:
   Probabilistic IO automata (PIOAs)
2. Communication model:
   General communication model where PIOAs communicate through buffers that need to be triggered to deliver a message. (No need to distinguish between real, ideal, and hybrid communication.)

Pfitzmann and Waidner 2001 (PIOA)

Computational model:

1. Computational entities:

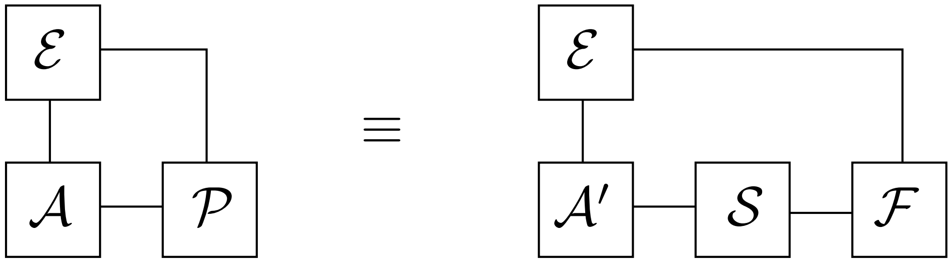
   Probabilistic IO automata (PIOAs)

2. Communication model:

   General communication model where PIOAs communicate through buffers that need to be triggered to deliver a message. (No need to distinguish between real, ideal, and hybrid communication.)

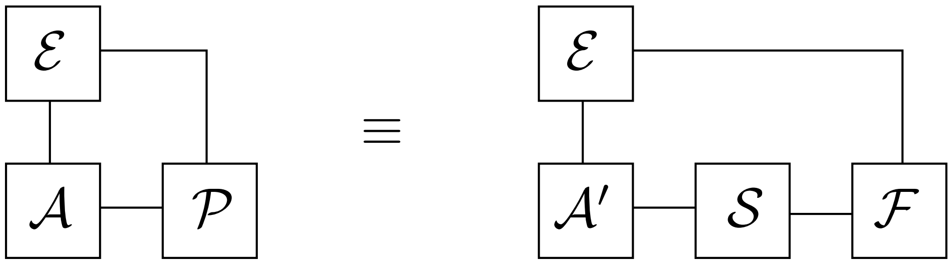
Security notions: UC + (strong) Black-box Simulatability (SBB).

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



Weak Black-box Simulatability (WBB)

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



Used in the literature to show UC (obviously: WBB implies UC).

Lincoln, Mitchell^2, Scedrov 1998 (PPC)

Computational model:

1. Computational entities:
   Probabilistic Polynomial-time Processes
2. Communication model:
   Probabilistic Process Calculus (PPC).

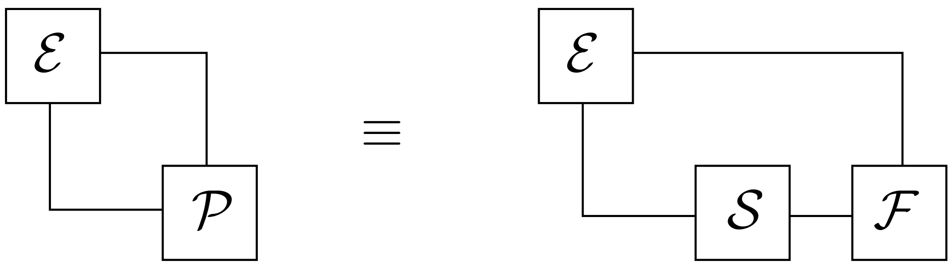
Lincoln, Mitchell^2, Scedrov 1998 (PPC)

Computational model:

1. Computational entities:
   Probabilistic Polynomial-time Processes
2. Communication model:
   Probabilistic Process Calculus (PPC).

Security notions: Process Congruence/Strong Simulatability (SS)

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

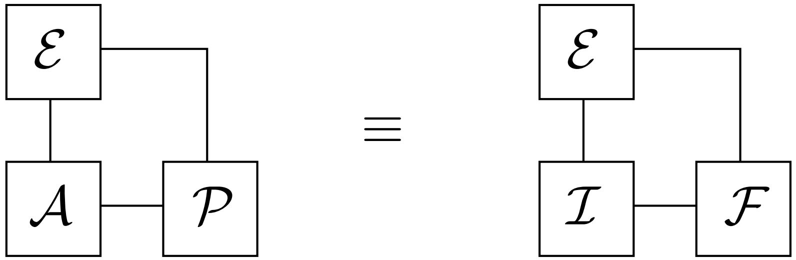


Even More Variety

Different variants of UC, BB, and SS have been considered!

UC

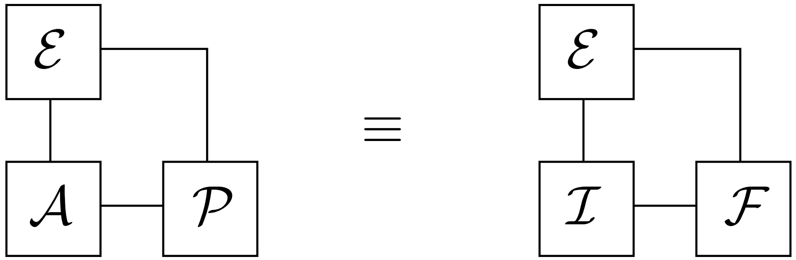
$\mathcal{P}$ and $\mathcal{F}$ are UC if $\forall\, \mathcal{A}\ \exists\, \mathcal{I}\ \forall\, \mathcal{E}$:



Distinguish between different tasks the processes perform:

UC

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

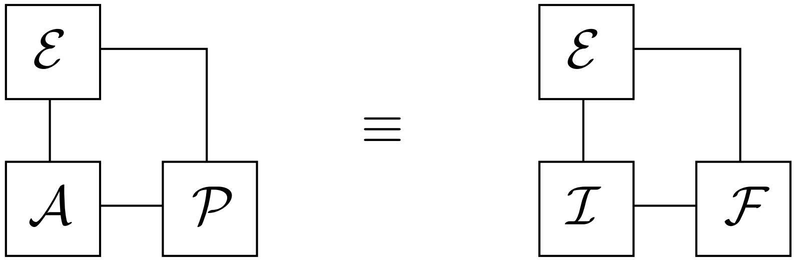


Distinguish between different tasks the processes perform:

Decision (distinguisher) process (D): May output a decision 1 or 0 depending on who the process believes to interact with. (environment)

UC

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



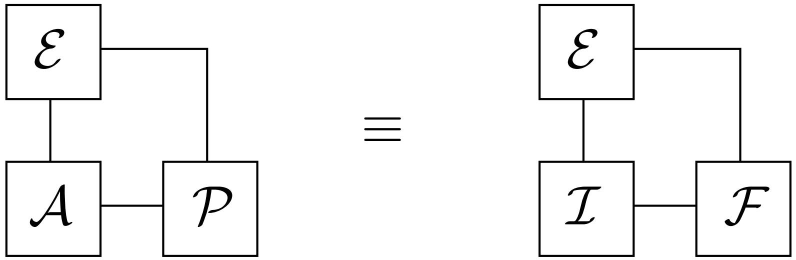
Distinguish between different tasks the processes perform:

Decision (distinguisher) process (D): May output a decision 1 or 0 depending on who the process believes to interact with. (environment)

Master process (M): Is triggered if no other process can go.

UC

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



Distinguish between different tasks the processes perform:

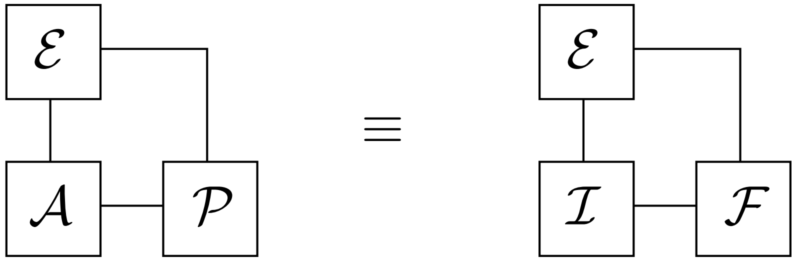
Decision (distinguisher) process (D): May output a decision 1 or 0 depending on who the process believes to interact with. (environment)

Master process (M): Is triggered if no other process can go.

Master decision process (MD): Is both master and decision process.

UC

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



Distinguish between different tasks the processes perform:

Decision (distinguisher) process (D): May output a decision 1 or 0 depending on who the process believes to interact with. (environment)

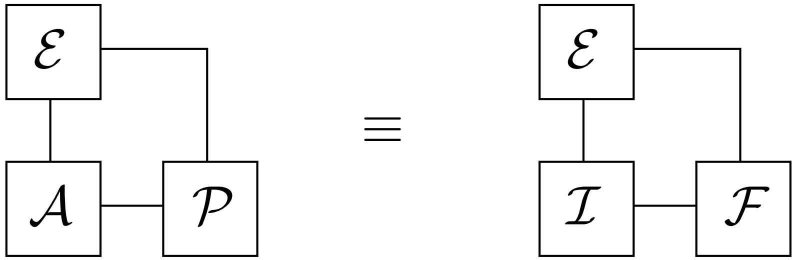
Master process (M): Is triggered if no other process can go.

Master decision process (MD): Is both master and decision process.

Regular process (R): Is neither a master nor a decision process. (e.g., real and ideal protocol)

UC

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



Distinguish between different tasks the processes perform:

Decision (distinguisher) process (D): May output a decision 1 or 0 depending on who the process believes to interact with. (environment)

Master process (M): Is triggered if no other process can go.

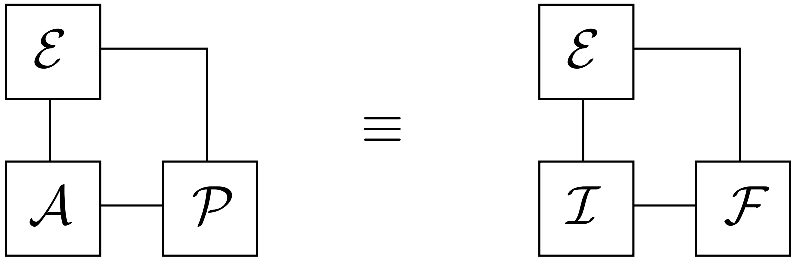
Master decision process (MD): Is both master and decision process.

Regular process (R): Is neither a master nor a decision process. (e.g., real and ideal protocol)

Who should be the master process?

UC

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

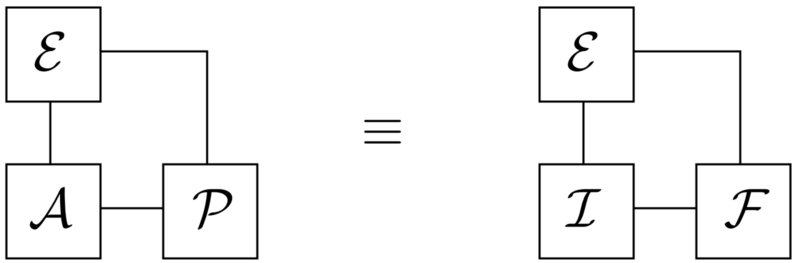


Literature provides different answers:

UC($\mathcal{A}$: **R**, $\mathcal{I}$: **R**, $\mathcal{E}$: **MD**) Canetti 2001

UC

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



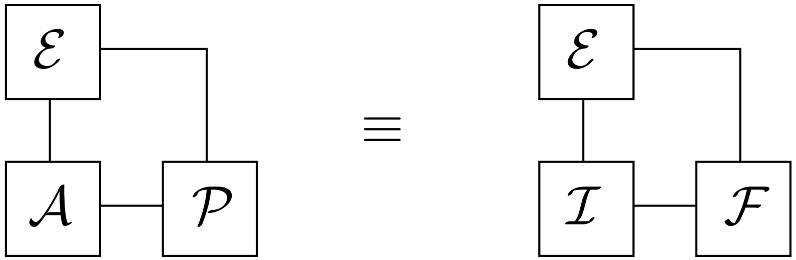
Literature provides different answers:

UC($\mathcal{A}$: **R**, $\mathcal{I}$: **R**, $\mathcal{E}$: **MD**) Canetti 2001

UC($\mathcal{A}$: **M**, $\mathcal{I}$: **M**, $\mathcal{E}$: **D**) Pfitzmann, Waidner 2001

UC

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

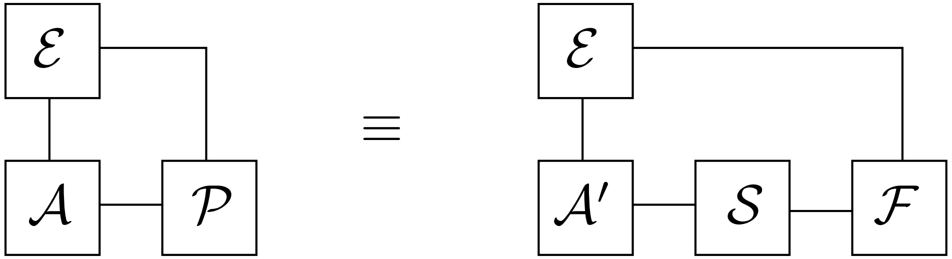


Literature provides different answers:

UC(	$\mathcal{A}$: **R**,	$\mathcal{I}$: **R**,	$\mathcal{E}$: **MD**	)	Canetti 2001
UC(	$\mathcal{A}$: **M**,	$\mathcal{I}$: **M**,	$\mathcal{E}$: **D**	)	Pfitzmann, Waidner 2001
UC(	$\mathcal{A}$: **M**,	$\mathcal{I}$: **M**,	$\mathcal{E}$: **MD**	)	Backes, Pfitzmann, Waidner 2004

SBB

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

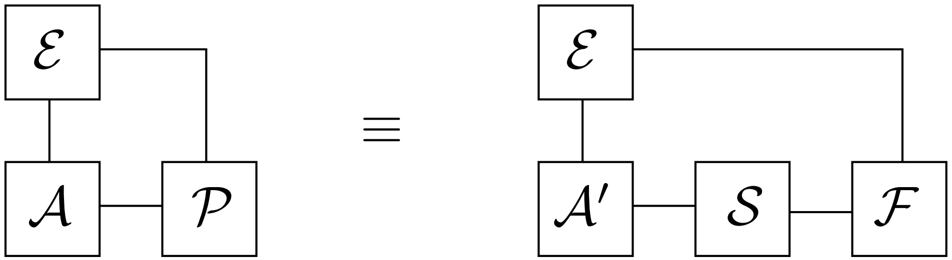


Variants:

SBB(	$\mathcal{A}$: **M**,	$\mathcal{S}$: **M**,	$\mathcal{E}$: **D**	) Pfitzmann, Waidner 2001
SBB(	$\mathcal{A}$: **M**,	$\mathcal{S}$: **M**,	$\mathcal{E}$: **MD**	) Backes, Pfitzmann, Waidner 2004
SBB(	$\mathcal{A}$: **M**,	$\mathcal{S}$: **R**,	$\mathcal{E}$: **MD**	)
SBB(	$\mathcal{A}$: **R**,	$\mathcal{S}$: **M**,	$\mathcal{E}$: **MD**	)
SBB(	$\mathcal{A}$: **R**,	$\mathcal{S}$: **R**,	$\mathcal{E}$: **MD**	)
SBB(	$\mathcal{A}$: **M**,	$\mathcal{S}$: **R**,	$\mathcal{E}$: **D**	)

Weak Black-box Simulatability (WBB)

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



Variants:

WBB($\mathcal{A}$: **M**, $\mathcal{S}$: **M**, $\mathcal{E}$: **MD**)

WBB($\mathcal{A}$: **M**, $\mathcal{S}$: **R**, $\mathcal{E}$: **MD**)

WBB($\mathcal{A}$: **R**, $\mathcal{S}$: **M**, $\mathcal{E}$: **MD**)

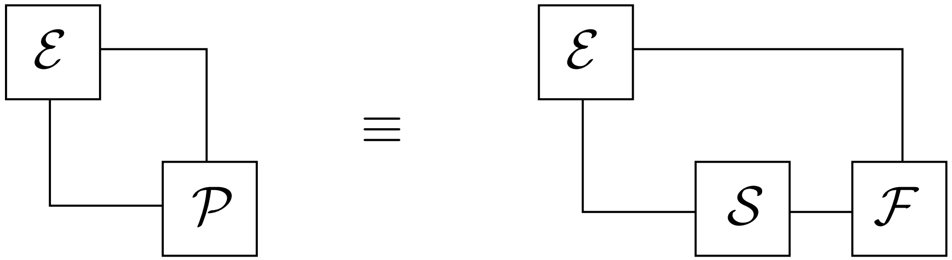
WBB($\mathcal{A}$: **R**, $\mathcal{S}$: **R**, $\mathcal{E}$: **MD**)

WBB($\mathcal{A}$: **M**, $\mathcal{S}$: **M**, $\mathcal{E}$: **D**)

WBB($\mathcal{A}$: **M**, $\mathcal{S}$: **R**, $\mathcal{E}$: **D**)

SS

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



Variants:

SS($\mathcal{S}$: **R**, $\mathcal{E}$: **MD**)

SS($\mathcal{S}$: **M**, $\mathcal{E}$: **MD**)

Relationship Between the Security Notions Across Models?

Relationship Between the Security Notions Across Models?

First, need general computational model that “subsumes” all other models.

Relationship Between the Security Notions Across Models?

First, need general computational model that “subsumes” all other models.

We introduce Sequential Probabilistic Process Calculus (SPPC).

Sequential Probabilistic Process Calculus (SPPC)

Syntactic and semantic restriction and extension of PPC.

Example process (simplified) corresponding to an IO automaton/ITM:

$$\mathcal{Q} = \quad !_{q(\mathbf{n})}\ \mathtt{in}(c_{\mathtt{s}}, x_s). \quad \sum_{c \in \mathcal{C}_{\mathtt{in}}} \mathtt{in}(c, x). \Bigg(\mathtt{out}(c_{\mathtt{ns}}, T_{ns}(c, x, x_s)) \mid\mid \sum_{c' \in \mathcal{C}_{\mathtt{out}}} \mathtt{in}(c_{\mathtt{ns}}, \langle x'_s, c', y \rangle). \Big(\mathtt{out}(c_{\mathtt{s}}, x'_s) \mid\mid \mathtt{out}(c', y) \Big) \Bigg)$$

Parallel composition of processes:

$$\mathcal{E} \mid\mid \mathcal{A} \mid\mid \mathcal{P}$$

Polynomial composition of processes (used in composition theorem):

$$\mathcal{E} \mid\mid \mathcal{A} \mid\mid !_{q(\mathbf{n})}\ \mathcal{P}$$

Important Feature of SPPC

Sequentiality (unlike PPC): Consider for instance $\mathcal{E} \mid\mid \mathcal{A} \mid\mid \mathcal{P}$.

1. At most one of the three processes is active.
2. The active process may send *at most one* message on an external channel *directly* to another process, and by reading the message, this other process is activated.

Important Feature of SPPC

Sequentiality (unlike PPC): Consider for instance $\mathcal{E} \,||\, \mathcal{A} \,||\, \mathcal{P}$.

1. At most one of the three processes is active.
2. The active process may send *at most one* message on an external channel *directly* to another process, and by reading the message, this other process is activated.

In comparison: PITM and PIOA are also sequential, but

PITM: Activation scheme is "hard-wired" into real, ideal, hybrid model.

PIOA: IO automaton may send *many* messages into different buffers (asynchronous network) and by triggering one buffer one message is delivered.

Advantage of SPPC

Simplicity: Details of network communication (buffers, specific triggering mechanisms, tapes) are not made explicit in SPPC, but

Flexibility: Are part of the protocol specification. For instance, all of the following can be modeled:

1. Insecure, authenticated, secure channels (with your favorite buffers, tapes,...)
2. Synchronous communication.
3. Broadcasting, etc.

Advantage of SPPC

Simplicity: Details of network communication (buffers, specific triggering mechanisms, tapes) are not made explicit in SPPC, but

Flexibility: Are part of the protocol specification. For instance, all of the following can be modeled:

1. Insecure, authenticated, secure channels (with your favorite buffers, tapes,...)
2. Synchronous communication.
3. Broadcasting, etc.

$\Longrightarrow$ SPPC allows to embed other models.

Our Results

Relationships between the security notions in SPPC:

Our Results

Relationships between the security notions in SPPC:

"Making the environment the master process unifies all notions."

Our Results

Relationships between the security notions in SPPC:

"Making the environment the master process unifies all notions."

More specifically, the following notions are equivalent:

1. UC($\mathcal{A}$: **R**, $\mathcal{I}$: **R**, $\mathcal{E}$: **MD**).
2. UC($\mathcal{A}$: **M**, $\mathcal{I}$: **M**, $\mathcal{E}$: **MD**).
3. WBB($\mathcal{A}$: **R**/**M**, $\mathcal{S}$: **R**/**M**, $\mathcal{E}$: **MD**).
4. All variants of SS and SBB (independent of whether $\mathcal{E}$ is **D** or **MD**).

Our Results

Relationships between the security notions in SPPC:

"Making the environment the master process unifies all notions."

More specifically, the following notions are equivalent:

1. UC($\mathcal{A}$: **R**, $\mathcal{I}$: **R**, $\mathcal{E}$: **MD**).
2. UC($\mathcal{A}$: **M**, $\mathcal{I}$: **M**, $\mathcal{E}$: **MD**).
3. WBB($\mathcal{A}$: **R**/**M**, $\mathcal{S}$: **R**/**M**, $\mathcal{E}$: **MD**).
4. All variants of SS and SBB (independent of whether $\mathcal{E}$ is **D** or **MD**).

Assuming the real protocol $\mathcal{P}$ is network predictable, i.e., it is possible to predict on what network channels $\mathcal{P}$ accepts messages depending on the traffic on the network channels.

Without this assumption, SS and SBB are stronger than the other two notions.

Our Results

Relationships between the security notions in SPPC:

UC($\mathcal{A}$: **R**, $\mathcal{I}$: **R**, $\mathcal{E}$: **MD**)
UC($\mathcal{A}$: **M**, $\mathcal{I}$: **M**, $\mathcal{E}$: **MD**)
WBB($\mathcal{A}$: **R**/**M**, $\mathcal{S}$: **R**/**M**, $\mathcal{E}$: **MD**)
and all variants of SS and SBB

$\Longrightarrow$ $\not\Longleftarrow$

UC($\mathcal{A}$: **M**, $\mathcal{I}$: **M**, $\mathcal{E}$: **D**)
WBB($\mathcal{A}$: **M**, $\mathcal{S}$: **M**, $\mathcal{E}$: **D**)

$\Searrow$ $\Nwarrow$? $\Nearrow$ $\not\Swarrow$

WBB($\mathcal{A}$: **M**, $\mathcal{S}$: **R**, $\mathcal{E}$: **D**)

Consequences for other models

PITM (Canetti 2001):

$$\text{UC}(\mathcal{A}\text{: }\mathbf{R},\ \mathcal{I}\text{: }\mathbf{R},\ \mathcal{E}\text{: }\mathbf{MD}) \iff \text{WBB}(\mathcal{A}\text{: }\mathbf{R},\ \mathcal{S}\text{: }\mathbf{R},\ \mathcal{E}\text{: }\mathbf{MD})$$

$$\approx \text{UC'}(\mathcal{A}\text{: }\mathbf{R},\ \mathcal{I}\text{: }\mathbf{R},\ \mathcal{E}\text{: }\mathbf{MD})$$

Consequences for other models

PIOA:

Pfitzmann, Waidner 2001:

$$\text{UC}(\mathcal{A}\colon \mathbf{M}, \mathcal{I}\colon \mathbf{M}, \mathcal{E}\colon \mathbf{D}) \quad \overset{\Longleftarrow}{\not\Longrightarrow} \quad \text{SBB}(\mathcal{A}\colon \mathbf{M}, \mathcal{S}\colon \mathbf{M}, \mathcal{E}\colon \mathbf{D})$$

Consequences for other models

PIOA:

Pfitzmann, Waidner 2001:

$$\mathrm{UC}(\mathcal{A}: \mathbf{M}, \mathcal{I}: \mathbf{M}, \mathcal{E}: \mathbf{D}) \quad \begin{array}{c}\Longleftarrow \\ \not\Longrightarrow\end{array} \quad \mathrm{SBB}(\mathcal{A}: \mathbf{M}, \mathcal{S}: \mathbf{M}, \mathcal{E}: \mathbf{D})$$

Backes, Pfitzmann, Waidner 2004:

$$\mathrm{UC}(\mathcal{A}: \mathbf{M}, \mathcal{I}: \mathbf{M}, \mathcal{E}: \mathbf{MD}) \quad \begin{array}{c}\Longleftarrow \\ \not\Longrightarrow\end{array} \quad \begin{array}{l}\mathrm{SBB}(\mathcal{A}: \mathbf{M}, \mathcal{S}: \mathbf{M}, \mathcal{E}: \mathbf{MD}) \\ \text{even if } \mathcal{P} \text{ is network predictable}\end{array}$$

Problem: Buffers and trigger mechanism used in PIOA.

Solution: Drop buffers and let IO automata talk to each other directly (similar to SPPC).

Results provide counterexamples for a theorem proved in Backes et al. 2004.

Correspondence Between PITM and PIOA Results

Embedding PITM into SPPC:

$$\mathrm{UC}_{PITM}(\mathcal{P},\mathcal{F}) \quad \text{iff} \quad \mathrm{UC}_{SPPC}(\mathrm{SPPC}(\mathcal{P}),\mathrm{SPPC}(\mathcal{F}))$$

Embedding PIOA* (PIOA without buffers) into SPPC:

$$\mathrm{SBB}_{PIOA^*}(\mathcal{P},\mathcal{F}) \quad \text{iff} \quad \mathrm{SBB}_{SPPC}(\mathrm{SPPC}(\mathcal{P}),\mathrm{SPPC}(\mathcal{F}))$$

Correspondence Between PITM and PIOA Results

Embedding PITM into SPPC:

$$\text{UC}_{PITM}(\mathcal{P},\mathcal{F}) \quad \text{iff} \quad \text{UC}_{SPPC}(\text{SPPC}(\mathcal{P}),\text{SPPC}(\mathcal{F}))$$

Embedding PIOA* (PIOA without buffers) into SPPC:

$$\text{SBB}_{PIOA^*}(\mathcal{P},\mathcal{F}) \quad \text{iff} \quad \text{SBB}_{SPPC}(\text{SPPC}(\mathcal{P}),\text{SPPC}(\mathcal{F}))$$

Equivalence: $\mathcal{P}_{PITM}$ (PITM) is equivalent to $\mathcal{P}_{PIOA^*}$ (PIOA*) iff

$$\text{SPPC}(\mathcal{P}_{PITM}) \cong \text{SPPC}(\mathcal{P}_{PIOA^*}),$$

i.e., $\mathcal{E} \mid\mid \text{SPPC}(\mathcal{P}_{PITM}) \equiv \mathcal{E} \mid\mid \text{SPPC}(\mathcal{P}_{PIOA^*}) \ \forall\, \mathcal{E}$.

Correspondence Between PITM and PIOA Results

Embedding PITM into SPPC:

$$\text{UC}_{PITM}(\mathcal{P},\mathcal{F}) \quad \text{iff} \quad \text{UC}_{SPPC}(\text{SPPC}(\mathcal{P}),\text{SPPC}(\mathcal{F}))$$

Embedding PIOA* (PIOA without buffers) into SPPC:

$$\text{SBB}_{PIOA^*}(\mathcal{P},\mathcal{F}) \quad \text{iff} \quad \text{SBB}_{SPPC}(\text{SPPC}(\mathcal{P}),\text{SPPC}(\mathcal{F}))$$

Equivalence: $\mathcal{P}_{PITM}$ (PITM) is equivalent to $\mathcal{P}_{PIOA^*}$ (PIOA*) iff

$$\text{SPPC}(\mathcal{P}_{PITM}) \cong \text{SPPC}(\mathcal{P}_{PIOA^*}),$$

i.e., $\mathcal{E} \,||\, \text{SPPC}(\mathcal{P}_{PITM}) \equiv \mathcal{E} \,||\, \text{SPPC}(\mathcal{P}_{PIOA^*}) \;\forall\, \mathcal{E}$.

Consequence of our results:

Given $\mathcal{P}_{PITM} \cong \mathcal{P}_{PIOA^*}$ and $\mathcal{F}_{PITM} \cong \mathcal{F}_{PIOA^*}$, we have:

$$\text{UC}_{PITM}(\mathcal{P}_{PITM},\mathcal{F}_{PITM}) \quad \text{iff} \quad \text{SBB}_{PIOA^*}(\mathcal{P}_{PIOA^*},\mathcal{F}_{PIOA^*})$$

Conclusion

- Introduced SPPC as a general computational model for simulation-based security notions that allows to embed other models.

  $\Longrightarrow$ Theorems proved in this model are valid for a broad class of other more specific models.

- Clarified the relationships between different security notions (UC, SBB, WBB, SS) and their variants as considered in the literature. Our proofs are based on a few equational principles.

  $\Longrightarrow$ "Making the environment the master process unifies all security notions."

  $\Longrightarrow$ With appropriate modifications (drop buffers in PIOA), results for SBB/UC proved in PIOA carry over to UC in PITM, and vice versa.

- Proved composition theorem for SPPC.

- Future work: Are there realistic attacks in a concurrent (non-sequential) framework (such as concurrent PPC) not captured by a sequential framework (such as SPPC, PIOA, PITM)?