COMPUTATIONAL INDISTINGUISHABILTY AND OBSERVATIONAL EQUIVALENCE

Hubert Comon-Lundh

(Joint work with Véronique Cortier)

h.comon-lundh@aist.go.jp

MOTIVATION

B: I want to sell my product. I am sure that the protocol is secure, however my client requires a proof. What should I do ?

R: Look, here is the definition of universal composability, simply show that your protocol satisfies the property

B: (after weighing up the document): can you do that for me ?

R: no way, it would take me 6 months and I have no time

B: 6 months ? But my product should be on the market next week !

How is it possible to deliver satisfactory proofs (or attacks) within short delays ?

SYMBOLIC VS. COMPUTATIONAL MODELS

	Computational model	Symbolic model
messages	bitstrings	abstract expressions
agents ‖ intruder	Network of Randomized Turing machines	process algebra
Attacker computation power	Any polynomial time computable function	A fixed set of function symbols
Security	Asymptotic, probabilistic: for any attacker, if the keys are large enough, it is very unlikely to break the security	Absolute: no probabilities, unconditional

The Challenge

Symbolic security $+$ Security of primitives $\Rightarrow$ Computational security

Split the problem and

Get the best of the two worlds !

COMPUTATIONAL INTERPRETATION AND PARSING

Comp. interpretation

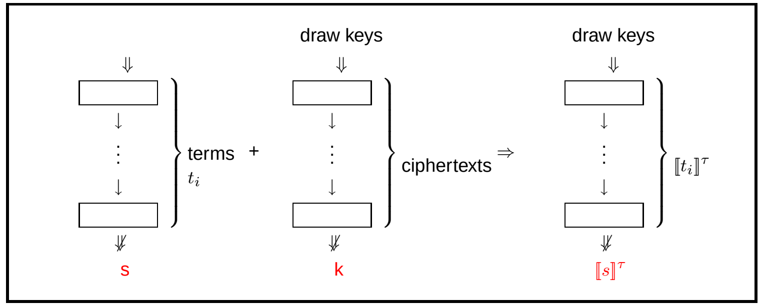
$\longrightarrow$

names $k, k_1, ...$	draw random numbers $\tau(k), \tau(k_1), \ldots$
function symbols: f, $\{\cdot\}^{\cdot}_{\cdot}$, $< \cdot, \cdot >, \ldots$	Actual PTIME computable functions $[\![f]\!], \mathcal{E}, \ldots$
terms $t, f(t_1, \ldots, t_n)$	$[\![f(t_1, \ldots, t_n)]\!]^\tau = [\![f]\!]([\![t_1]\!]^\tau, \ldots, [\![t_n]\!]^\tau)$
predicate symbols: p, typing, =,...	$[\![p]\!]([\![t_1]\!]^\tau, \ldots [\![t_n]\!]^\tau) = p(t_1, ..., t_n)$ with overwhelming probability
(Simple) Processes	Communicating Turing machines

$\longleftarrow$

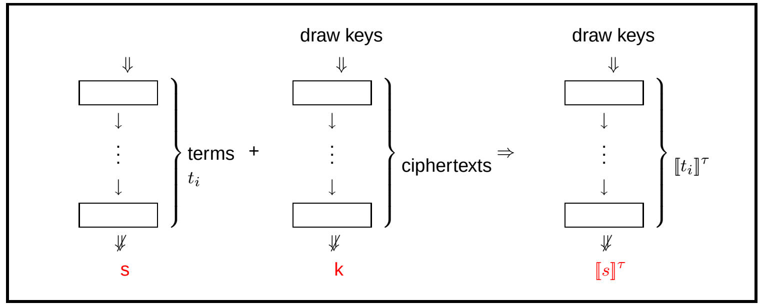
parsing

WHICH NOTIONS OF SECURITY ?

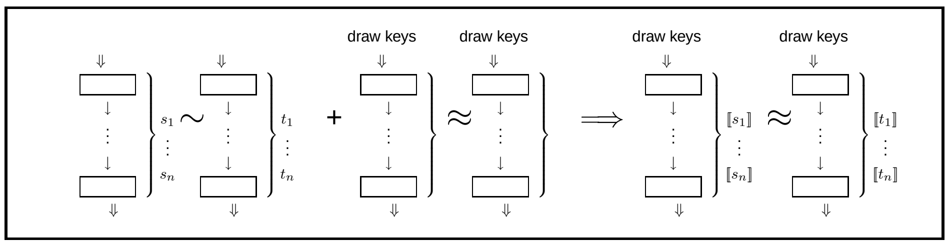


Passive attacks & reachability property

WHICH NOTIONS OF SECURITY ?

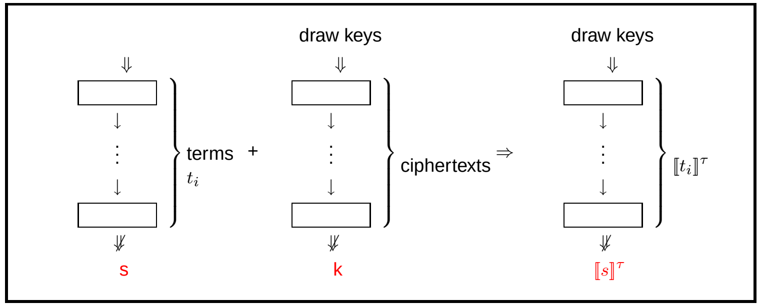


Passive attacks & reachability property

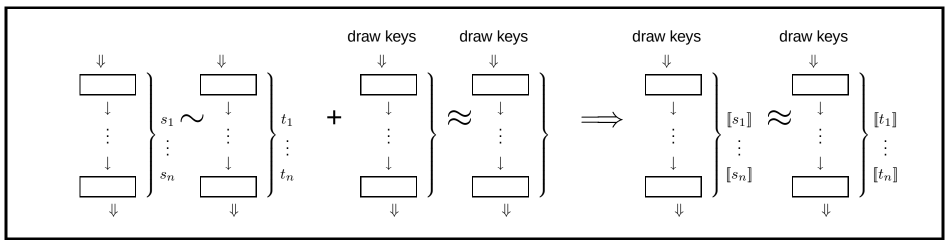


Passive attacks, indistinguishability [Abadi & Rogaway 2000 ...]

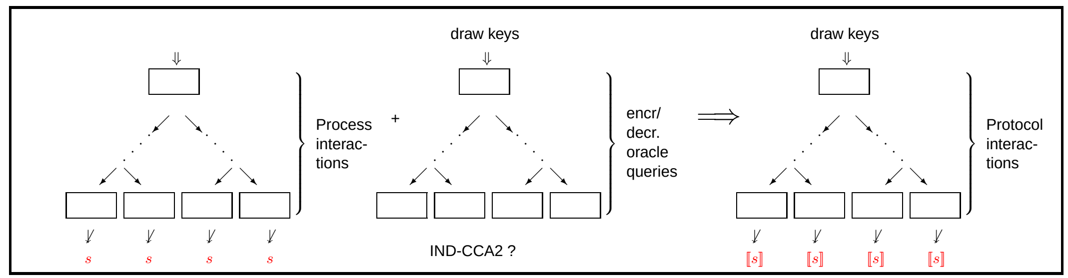
WHICH NOTIONS OF SECURITY ?



Passive attacks & reachability property

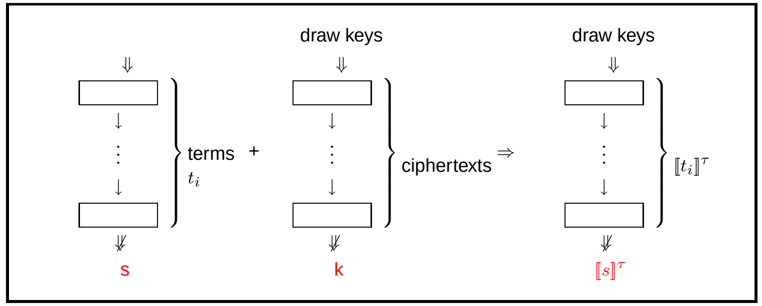


Passive attacks, indistinguishability [Abadi & Rogaway 2000 ...]

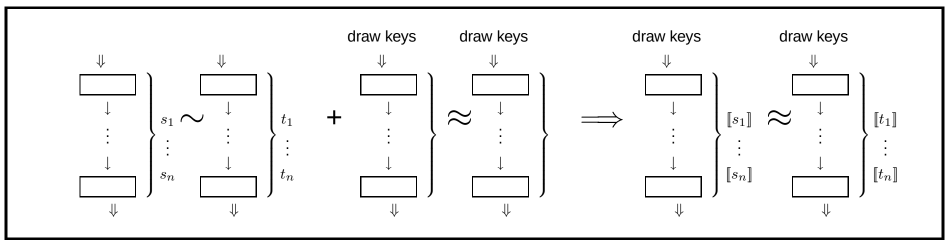


Active attacks & Reachability properties [Backes,Pfitzmann 2003; Cortier, Warinschi 2005,...]

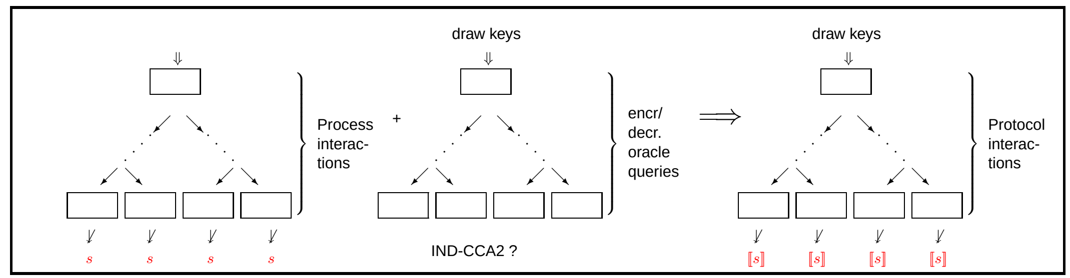
WHICH NOTIONS OF SECURITY ?



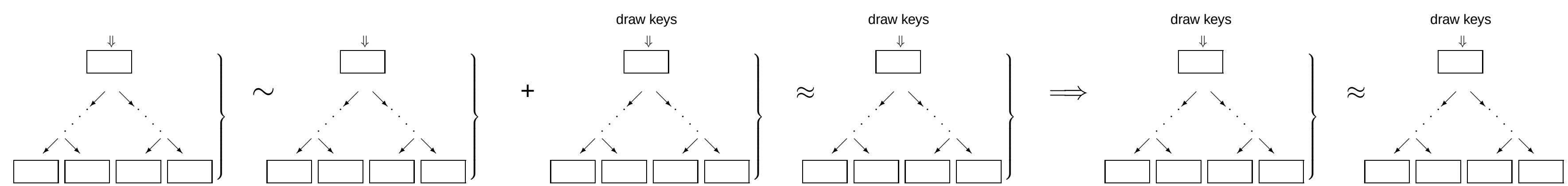
Passive attacks & reachability property



Passive attacks, indistinguishability [Abadi & Rogaway 2000 ...]



Active attacks & Reachability properties [Backes,Pfitzmann 2003; Cortier, Warinschi 2005,...]



Observational equivalence

Security of primitives: IND-CPA, INT-CTXT...

Active attacks & Indistinguishability; Universal composability

THE CHALLENGE (NEW FORMULATION)

Universal composability vs. Observational equivalence

THE CHALLENGE (NEW FORMULATION)

Universal composability vs. Observational equivalence

In any environment, an attacker should not be able to distinguish between two distributed programs

vs.

THE CHALLENGE (NEW FORMULATION)

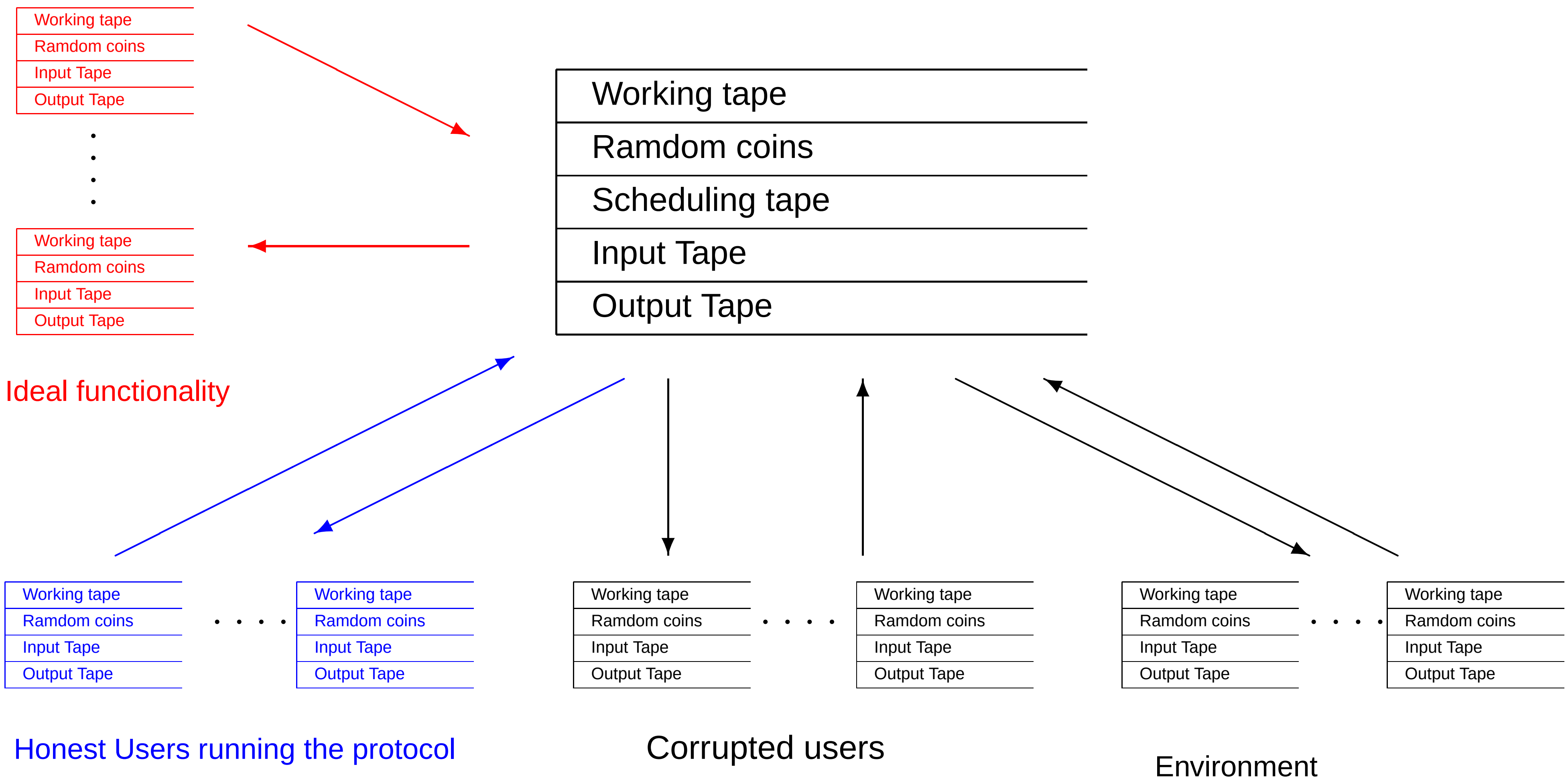
Universal composability vs. Observational equivalence

In any environment, an attacker should not be able to distinguish between two distributed programs

vs.

In any environment, an attacker should not be able to distinguish between two distributed programs

Interactive Turing Machines



All machines run in polynomial time

COMPUTATIONAL INDISTINGUISHABILITY

$\mathcal{F}, \mathcal{F}'$ are two networks of machines (including corrupted ones). $\mathcal{A}$ an interactive Turing machine restricted to work in probabilistic polynomial time.

$$\forall P, \forall \mathcal{A}, \exists N, \forall \eta > N.$$
$$|\mathbf{Pr}\{\overline{r}, r : (\mathcal{A}(r) \| \mathcal{F}(\overline{r}))(0^\eta) = 1\} - \mathbf{Pr}\{\overline{r}, r : (\mathcal{A}(r) \| \mathcal{F}'(\overline{r}))(0^\eta) = 1\}| < \frac{1}{P(\eta)}$$

Typical examples of security properties, which are not trace properties:

- “real or random”: $\mathcal{F}'$ is identical to $\mathcal{F}$ except that confidential value(s) shared by honest agents are replaced by random numbers.
- Anonimity
- guessing attacks
- ...

THE APPLIED π-CALCULUS

$Processes(P, Q)$	$::=$	$P \| Q$	Parallel composition
		$P!$	replication
		$\nu n.P$	n is a new name: scoping
		$c(x).P$	
		$\overline{c}(M) \cdot P$	Terms M build with a fixed set of primitives
		if C then P else Q	

Comm $\quad c(x) \cdot P \| \overline{c}(M) \cdot Q \quad \rightarrow \quad P\{x \mapsto M\} \| Q$

Computation via term rewriting, e.g. dec$($enc$(x, k), k) \rightarrow x$.
Rules for name scoping
Observational equivalence:

$$P \sim_o Q \quad \Longleftrightarrow \quad \forall I.((P \| I) \downarrow_c \Leftrightarrow (Q \| I) \downarrow_c)$$

SIMPLE PROCESSES

$$\nu n_1, \ldots n_k . B_1 \| \cdots \| B_n$$

Each B_i:

$$B ::= \forall \overline{x}.\nu \overline{n}.c(x) \cdot \text{ if } C \text{ then } \overline{c}(M) \cdot B' \text{ else } \mathbf{0}$$

Computational interpretation $[\![B]\!]$:

1. Draw the random values
2. Wait for input: this binds the variable x
3. check the condition (using the computational interpretations and bindings)
4. Send $[\![M]\!]$
5. $\cdots$

A SOUNDNESS RESULT

We assume a *key hierarchy*, a *parsing algorithm*, symmetric keys.

Theorem: If the encryption scheme is joint IND-CPA and INT-CTXT then, for any simple processes P, Q,

$$P \sim_o Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

Definition [Bellare & Namprempre, ASIACRYPT 2000] A symmetric encryption scheme is INT-CTXT if, for any non-null polynomial P with positive integer coefficients, for every PPT A with one oracle, there is a N such that, for all $\eta > N$,

$$\mathbf{Pr}\{k \stackrel{R}{\longleftarrow} \mathcal{K}_G(\eta), r, \overline{r} \stackrel{R}{\longleftarrow} U : \exists x, r'. A^{\mathcal{O}_k}(0^\eta \mid r) = \{x\}_k^{r'} \wedge x \notin S\} \leq \frac{1}{P(\eta)}$$

where $S = \{x_1, \ldots, x_n\}$ is the sequence of requests to the oracle, $\mathcal{O}_k(x_i) = \{x_i\}_k^{r_i}$ and $\overline{r}$ is the sequence $(r_1, \ldots, r_n)$.

MAIN INGREDIENTS OF THE PROOF

$$P \sim_o Q \Rightarrow T_P \sim T_Q \Rightarrow T_P \approx T_Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

MAIN INGREDIENTS OF THE PROOF

$$P \sim_o Q \Rightarrow T_P \sim T_Q \Rightarrow T_P \approx T_Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

$P \sim_o Q \Rightarrow T_P \sim T_Q$: A process algebra result in the spirit of characterization of observational equivalence by labeled bisimilarity

MAIN INGREDIENTS OF THE PROOF

$$P \sim_o Q \Rightarrow T_P \sim T_Q \Rightarrow T_P \approx T_Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

$P \sim_o Q \Rightarrow T_P \sim T_Q$: A process algebra result in the spirit of characterization of observational equivalence by labeled bisimilarity

$T_P \sim T_Q \Rightarrow T_P \approx T_Q$: uses the (tree) soundness in the ground case. This is a new concept, which generalizes the soundness of static equivalence from sequences to trees. It is necessary for the preservation of trace equivalences.

MAIN INGREDIENTS OF THE PROOF

$$P \sim_o Q \Rightarrow T_P \sim T_Q \Rightarrow T_P \approx T_Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$$

$P \sim_o Q \Rightarrow T_P \sim T_Q$: A process algebra result in the spirit of characterization of observational equivalence by labeled bisimilarity

$T_P \sim T_Q \Rightarrow T_P \approx T_Q$: uses the (tree) soundness in the ground case. This is a new concept, which generalizes the soundness of static equivalence from sequences to trees. It is necessary for the preservation of trace equivalences.

$T_P \approx T_Q \Rightarrow [\![P]\!] \approx [\![Q]\!]$ any computational trace is, with an overwhelming probability, an instance of a symbolic trace.

KEY LEMMAS

$$
\begin{array}{rcll}
\Psi_k(n) & = & n & \text{if } n \text{ is a name or a constant} \\
\Psi_k(< t_1, t_2 >) & = & < \Psi_k(t_1), \Psi_k(t_2) > & \\
\Psi_k(\{t\}_k^r) & = & \{0^{l(u)}\}_k^r & \\
\Psi_k(\{t\}_{k'}^r) & = & \{\Psi_k(t)\}_{k'}^r & \text{if } k \neq k'
\end{array}
$$

Ψ_k is extended to computation trees: the adversary is given a view of the execution where any encryption by k has been replaced by encryption of zeros by k.

Lemma: For any computation tree T and for any name k such that k is not deducible from T, $T \sim \Psi_k(T)$.

Lemma: Let P_1 and P_2 be two simple processes such that each P_i admits a key ordering. Let T_{P_i} be the process computation tree associated to P_i. If $T_{P_1} \sim T_{P_2}$ then $T_{P_1} \approx T_{P_2}$ or the encryption scheme is not joint IND-CPA and INT-CTXT.

OPEN QUESTIONS AND FUTURE WORK

Easy:

- Only symmetric encryption –> more primitives
- Replication vs. arbitrary number of processes

Hard:

- Key cycles and key hierarchy
- No dynamic key disclosure (see also [Backes & Pfitzmann 2004])
- Forged keys must be requested to a certification authority (see also [Canetti & Herzog 2006])

Prospective:

- Worst case vs average case attacker
- Getting away from the asymptotic security