On Round-Optimal Zero Knowledge in the Bare Public Key Model

Alessandra Scafuro and Ivan Visconti
University of Salerno
ITALY

FOCUS: Round-Optimal (4 rounds)
concurrent and resettable Zero Knowledge
in the Bare Public Key Model

have already been achieved:

Round-optimal Concurrent ZK:
(standard assumptions)

- [Z03] only sequential soundness,
- [DV05] concurrent soundness,
- [V06] efficiently,
- [D09] minimal assumptions,
- [YZ10] sophisticated notion of argument of knowledge.

Round-optimal Resettable ZK:
(complexity leveraging)

- [MR01] only sequential soundness,
- [DPV04] concurrent soundness,
- [YZ07] under generic assumptions.

What do we do in this paper ?

Our Contribution

- Point-out a subtle issue in the zero knowledge proof of *all round-optimal* (concurrent and resettable) protocols.

  Alternative proof?

  Protocol's structure of almost all round-optimal protocols makes problematic the design of any simulator.

  Exceptions: *could* admit alternative simulators:

  - Resettable ZK of [YZ07]: uses complexity leveraging.
  - Concurrent ZK of [Z03]: only sequential soundness.

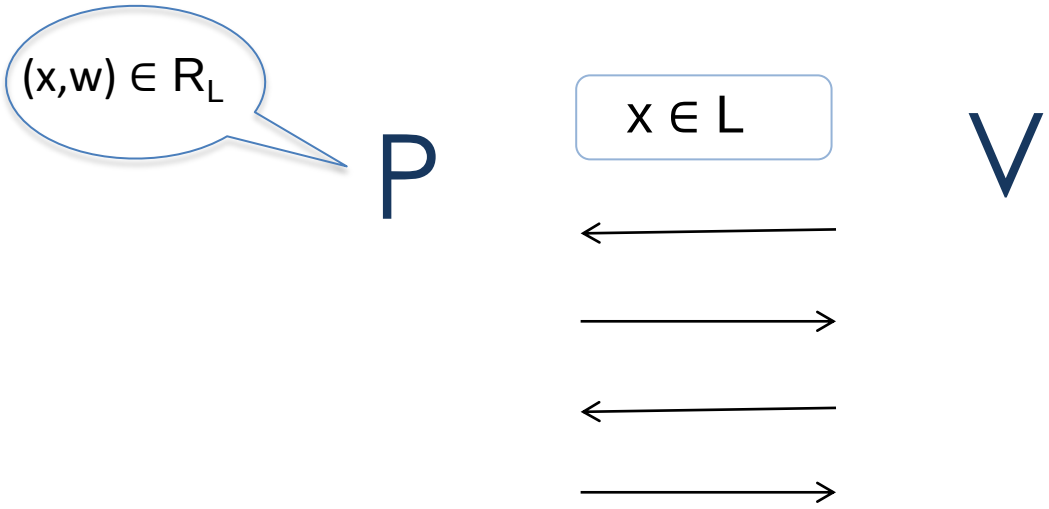
- New round-optimal concurrent ZK with concurrent soundness and standard assumptions.

  - The same protocol admits efficient implementation.
  - Round-optimal resettable ZK (similar to [YZ07]), with a new proof.

Outline

- Definitions
  - Concurrent Zero Knowledge
  - Bare Public Key (BPK) Model
  - Concurrent Zero Knowledge and Soundness in the BPK model
- Round-optimal Concurrent Zero Knowledge:
  - the issue of *all* zero-knowledge simulators
  - the difficulty of designing any alternative simulator
- Our technique

Zero knowledge Interactive Proofs
(standard model)



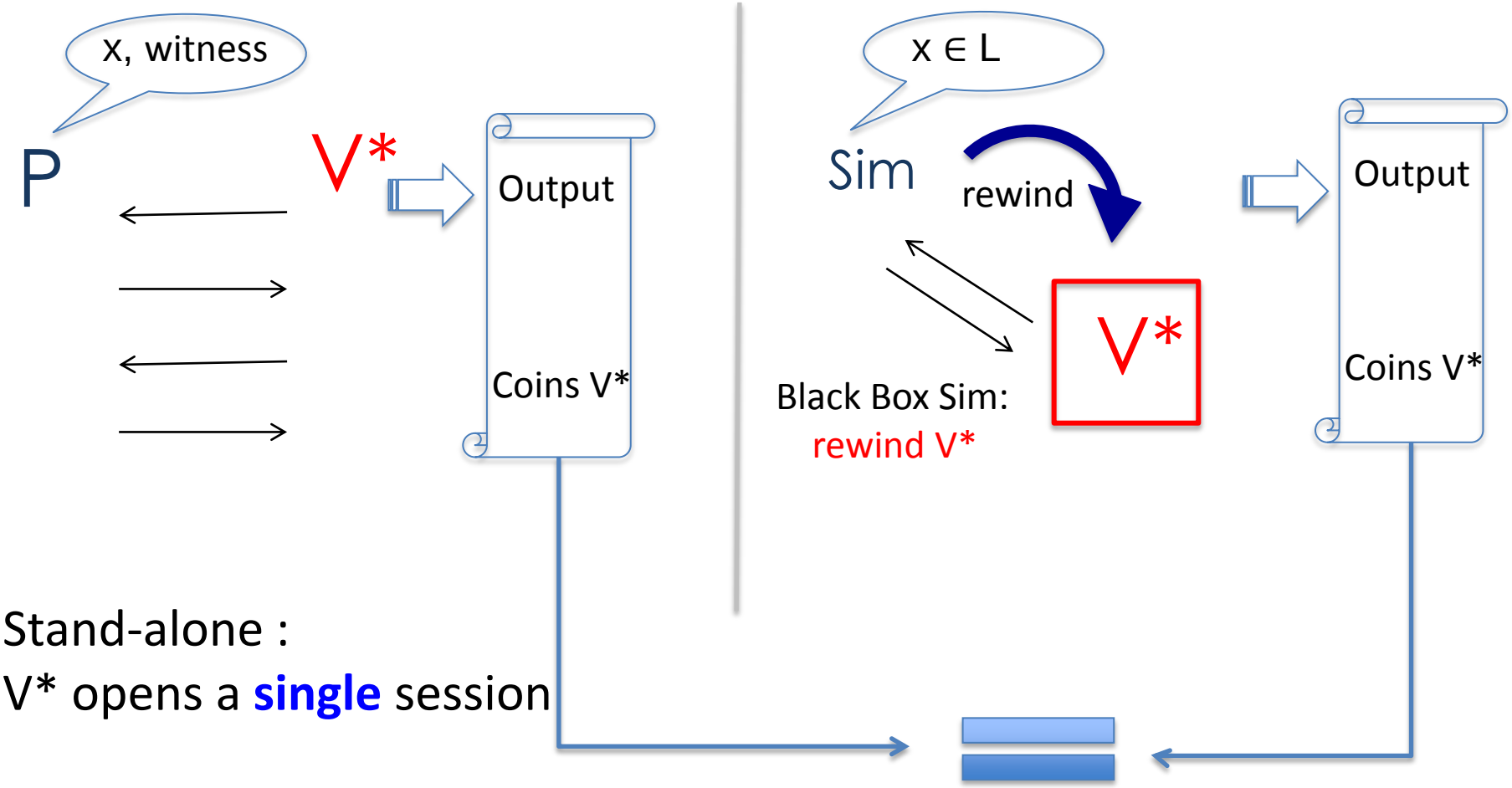
Completeness: if both P and V are honest, V accepts the proof.

Soundness: if the theorem is false any P* cannot convince V.

Zero Knowledge: (intuition) any V* learns nothing but the fact that the theorem is true.

Zero Knowledge (stand-alone)

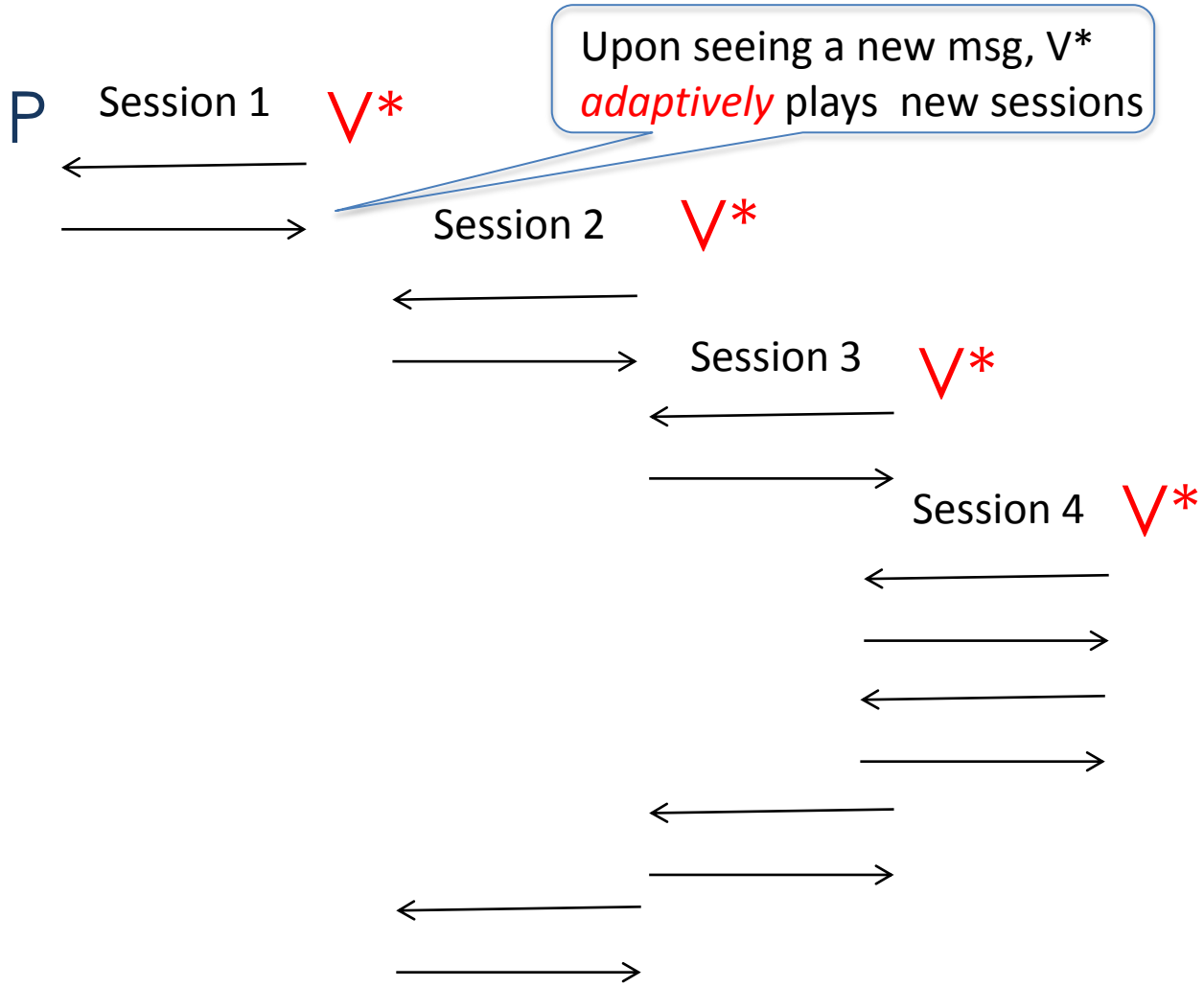
V* does not learn anything?



Stand-alone :
V* opens a **single** session

Concurrent Zero Knowledge

More realistic setting: V* can open *many sessions concurrently*.



Constant-round concurrent black-box Zero Knowledge (cZK) in the standard model is **impossible** [CKPR01].

Achieving black-box constant-round cZK *requires* setup assumptions.

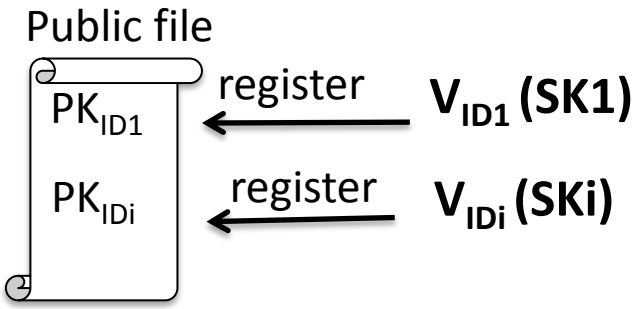
Bare Public Key Model

Introduced in STOC 2000 by Canetti, Goldreich, Goldwasser, Micali

Assumption: each verifier must be associated with a permanent public key, registered **before** any proof starts.

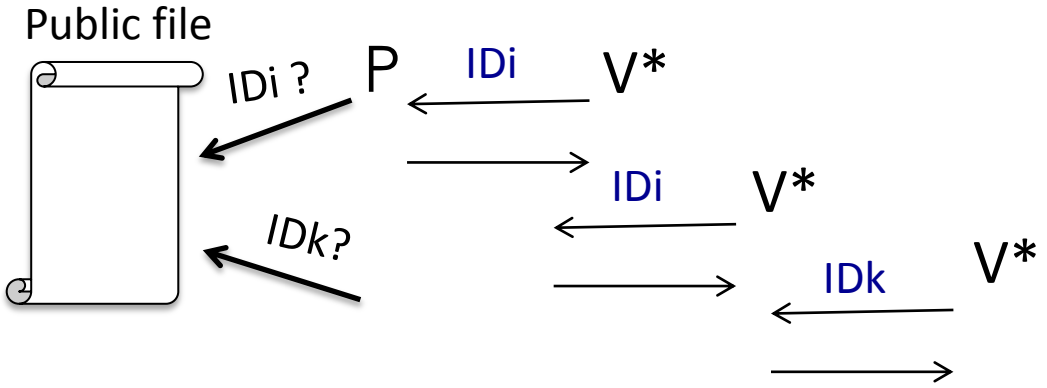
Registration Phase

- Non-interactive
- Fully controlled by V*
- No trusted party involved

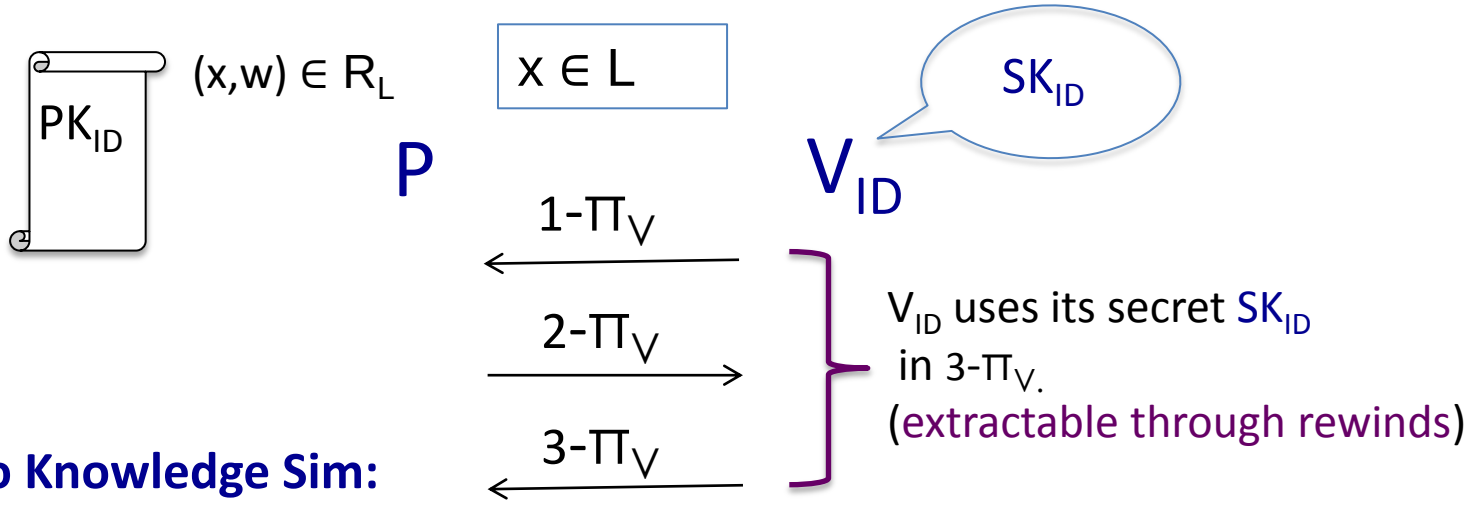


Proof Phase

- V* can still open an *unbounded* (poly) number of sessions.
- V* has full control of the schedule
- Restriction: V* cannot play with identity not in public file.



Achieving constant-round concurrent ZK in the BPK model



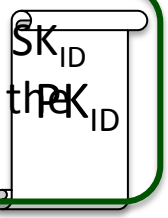
Concurrent Zero Knowledge Sim:

- gets SK_{ID} by rewinding Π_V
- runs Π_P in straight-line using SK_{ID}
- once SK_{ID} is extracted, all sessions played with V_{ID} are run in straight-line
- poly: number of extraction bounded by number of identities.

"is able to compute something computable only with knowledge of SK_{ID} "

Concurrent Soundness in the BPK model

IDEA: if known, the secret SK_{ID} should be used *already* in the first msg $1\text{-}\Pi_P$.



P*

V_{ID}

SK_{ID}

$1\text{-}\Pi_V$

$2\text{-}\Pi_V$

$3\text{-}\Pi_V$

Proving concurrent soundness: rule out MiM Attack

Concurrent executions

SK_{ID} V_{ID}

$1\text{-}\Pi_V$

(SK_{ID}, Π_P) $1\text{-}\Pi_P$

MiM

$2\text{-}\Pi_P$

$2\text{-}\Pi_V$

$3\text{-}\Pi_V$

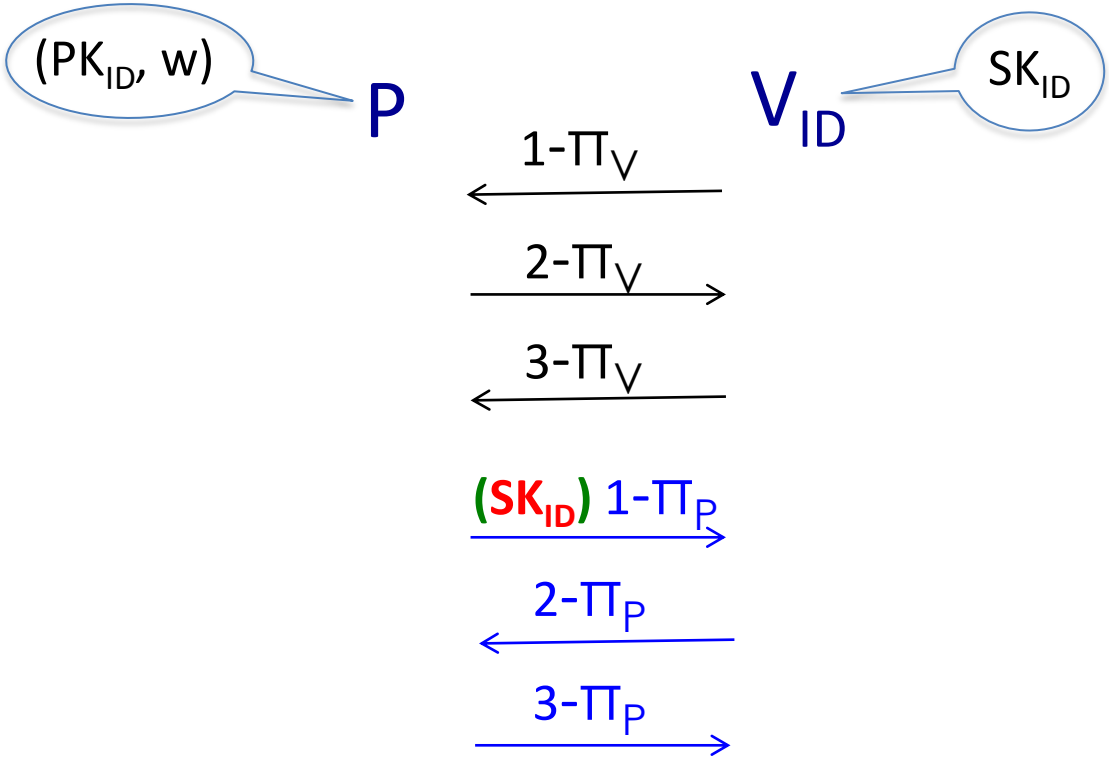
$3\text{-}\Pi_P$

P convinces V_{ID} if
1) it knows witness
OR
2) it knows SK_{ID}

Concurrent Zero Knowledge

Still preserved. Sim extracts the secret before having to play the first msg $1\text{-}\Pi_P$.

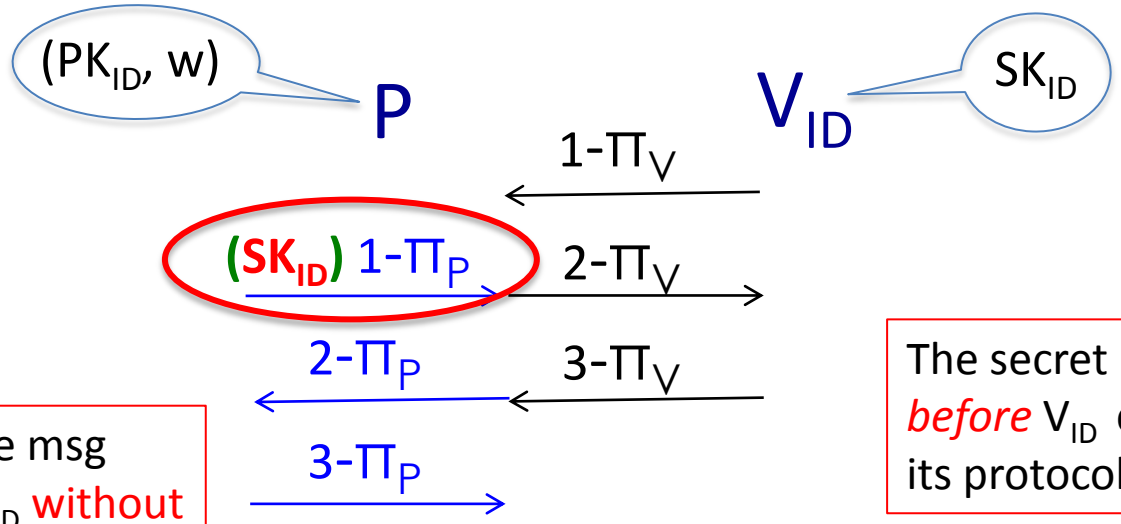
Concurrent Zero Knowledge and Soundness



Outline

- Definitions
  - Concurrent Zero Knowledge
  - Bare Public Key (BPK) Model
  - Concurrent Zero Knowledge and Soundness in the BPK model
- Round-optimal Concurrent Zero Knowledge:
  - the issue of *all* zero-knowledge simulators
  - the difficulty of designing any alternative simulator
- Our technique

Round-Optimal (4 rounds) Concurrent Zero Knowledge and Soundness



Sim has to play the msg dependent on SK_{ID} without knowing it yet.

The secret is used *before* V_{ID} completes its protocol.

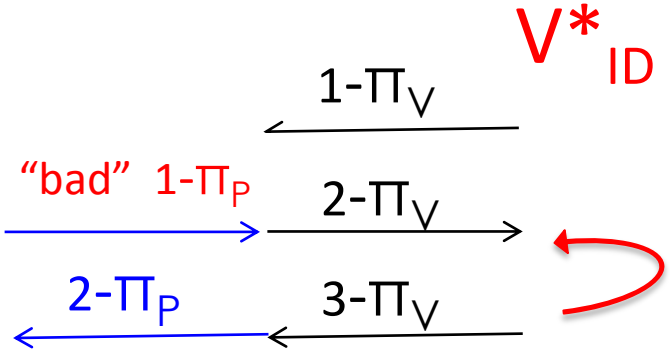
Concurrent Simulator?

Concurrent Simulator in Literature

all (*published*) simulators follow this strategy.

Simulation in phases

When playing with an "unresolved" identity: **Sim**



1) Play a "bad" first message

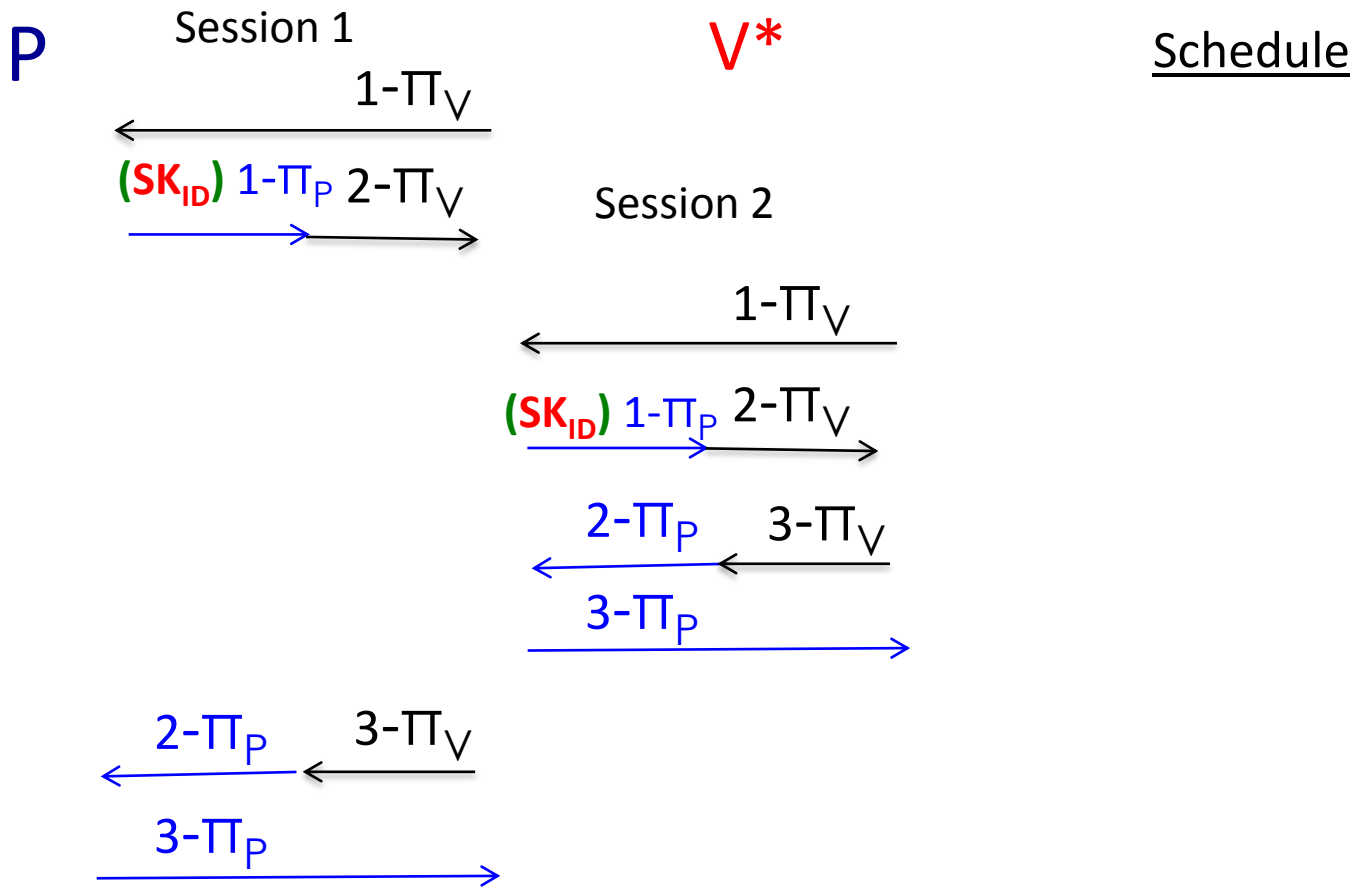
2) Extract the secret needed to solve the session.

3) Start simulation *from scratch* (a new phase) with knowledge of one more secret SK_{ID}.

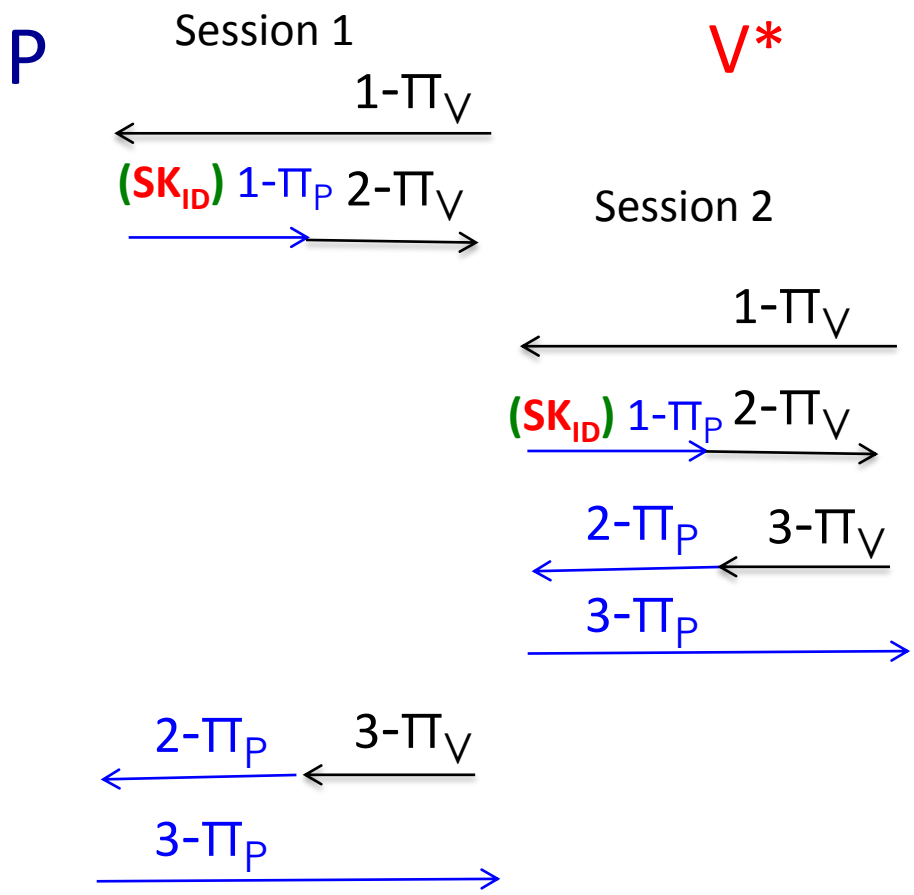
Number of phases = number of identities (poly)

Our contribution:
Such simulation approach leads to a distinguishable distribution.

A dummy attack



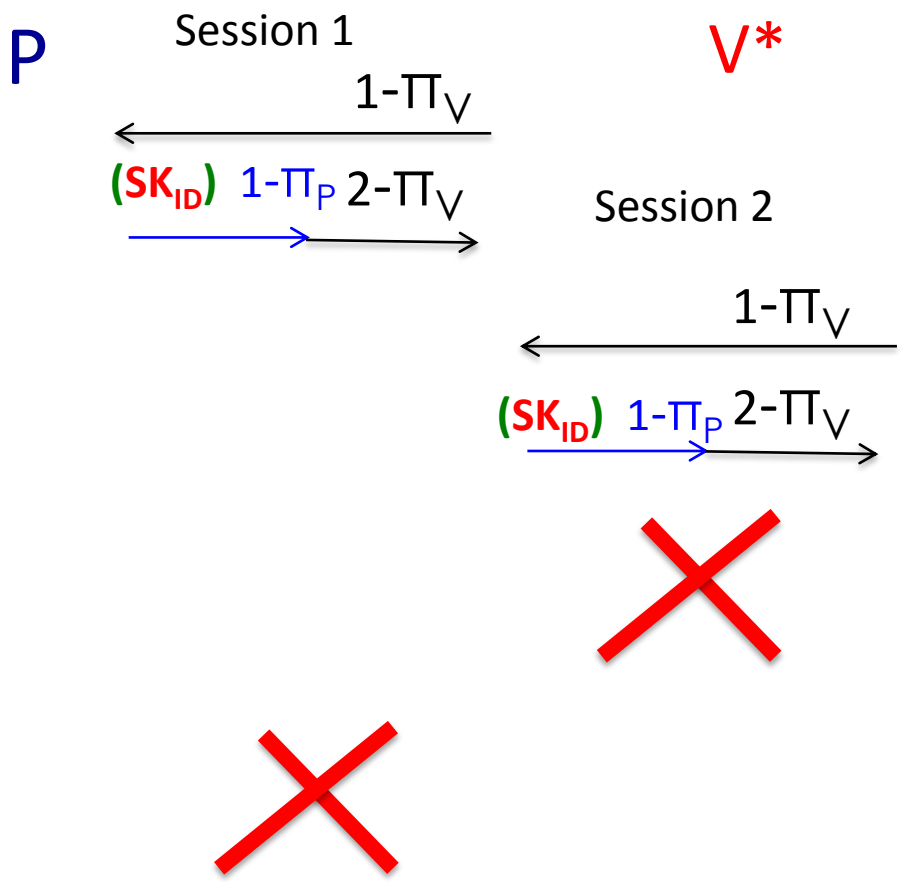
A dummy attack



V* Strategy

V* aborts Session 1 with prob. 1/2
V* aborts Session 2 with prob. 1/2
(taken over the transcript seen so far)

A dummy attack



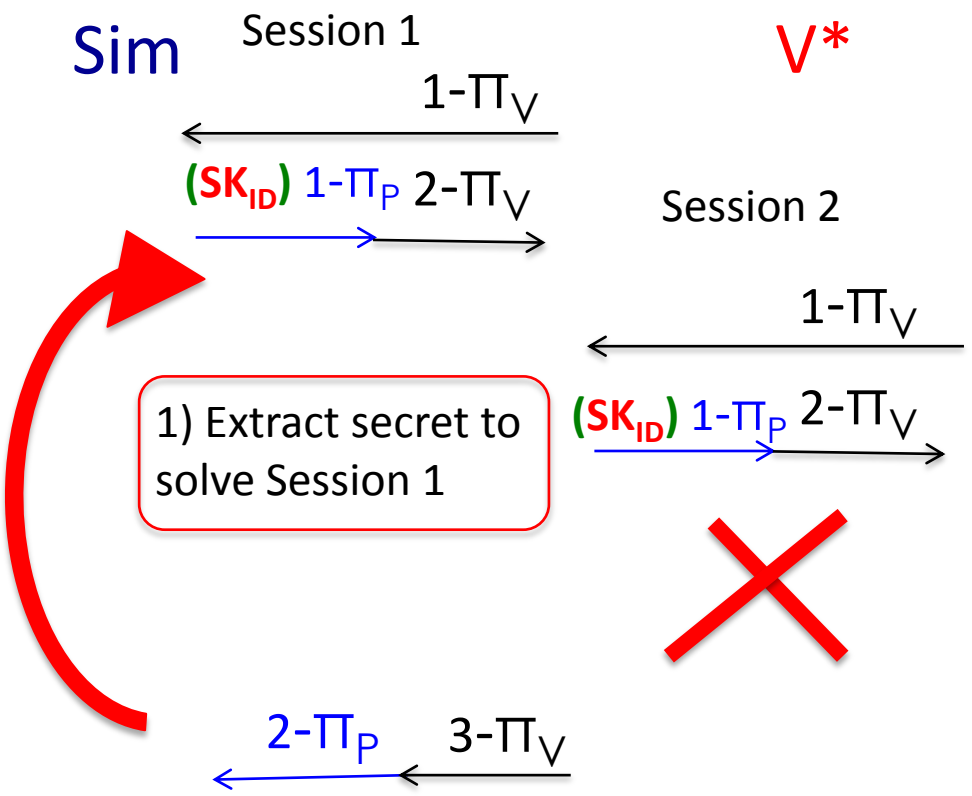
V* Strategy

V* aborts Session 1 with prob. 1/2
V* aborts Session 2 with prob. 1/2
(taken over the transcript seen so far)

Prob. Abort in Real Game

Pr [Abort S1] x Pr[Abort S2] =
1/2 x 1/2 = **1/4**

A dummy attack



V* Strategy

V* aborts Session 1 with prob. 1/2
V* aborts Session 2 with prob. 1/2
(taken over the transcript seen so far)

Prob. Abort in Real Game

Pr [Abort S1] x Pr[Abort S2] =
1/2 x 1/2 = **1/4**

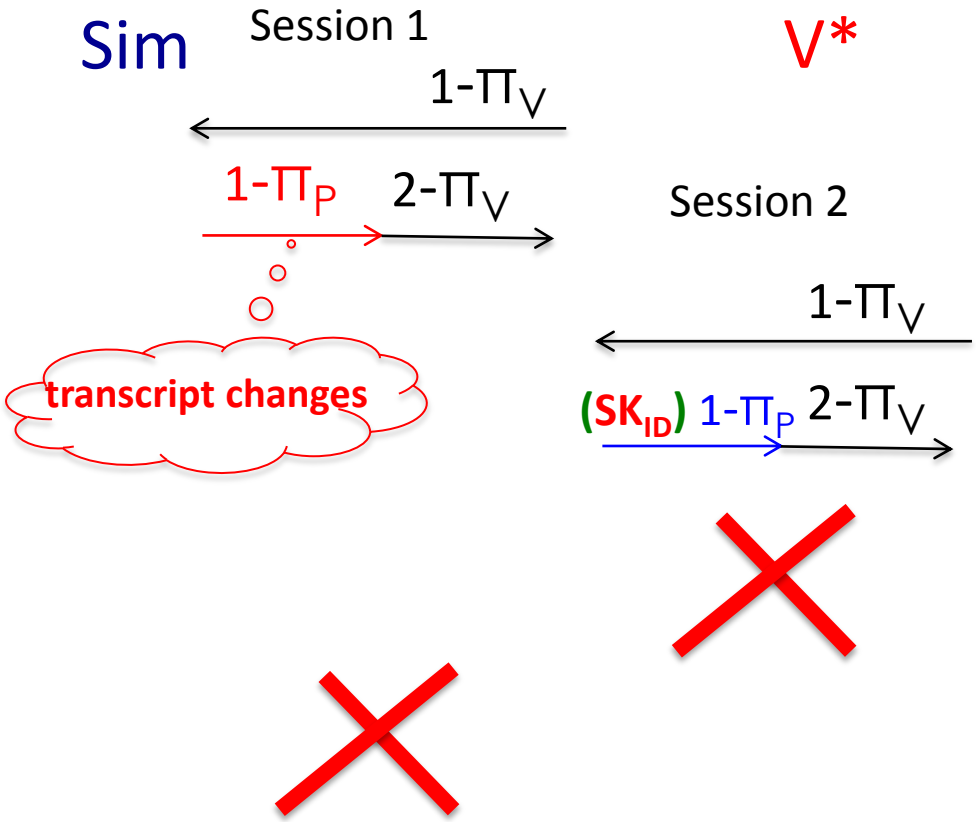
Prob. Abort Simulation

Case 1.
Pr [Abort S1] x Pr[Abort S2] =
1/2 x 1/2 = 1/4

Case 2.
Pr[Abort S2] x Pr[NOT Abort S1]

A dummy attack

2) Start the simulation from scratch with knowledge of secret.



Sim outputs two aborts with probability at least

Case 1 + Case 2 > Real Game

V* Strategy

V* aborts Session 1 with prob. 1/2
V* aborts Session 2 with prob. 1/2
(taken over the transcript seen so far)

Prob. Abort in Real Game

Pr [Abort S1] x Pr[Abort S2] =
1/2 x 1/2 = **1/4**

Prob. Abort Simulation

Case 1.
Pr [Abort S1] x Pr[Abort S2] =
1/2 x 1/2 = 1/4

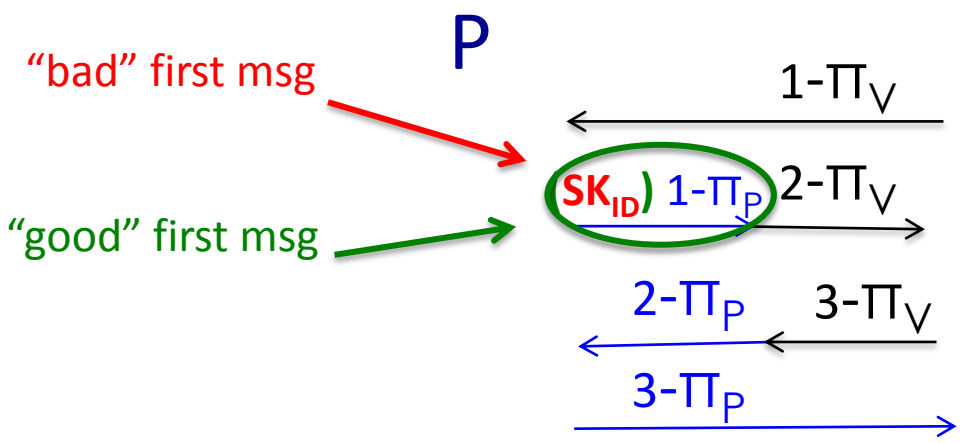
Case 2.
Pr[Abort S2] x Pr[NOT Abort S1] **x Pr[Case 1]**
= 1/2 x 1/2 x **1/4** = 1/16

Simulation in phases yields a distinguishable output.

Alternative Simulation Strategies?

- Trivially, there exists a simulator for the dummy V* seen so far.
- what about more sophisticated V* that aborts with different probability in different sessions….?

The problem: the protocol's structure of *round-optimal* protocols



Remark

Protocols that do not follow this structure could admit alternative strategies:

- resZK [YZ07] complexity leveraging.
- cZK [Z03]: only sequential soundness.

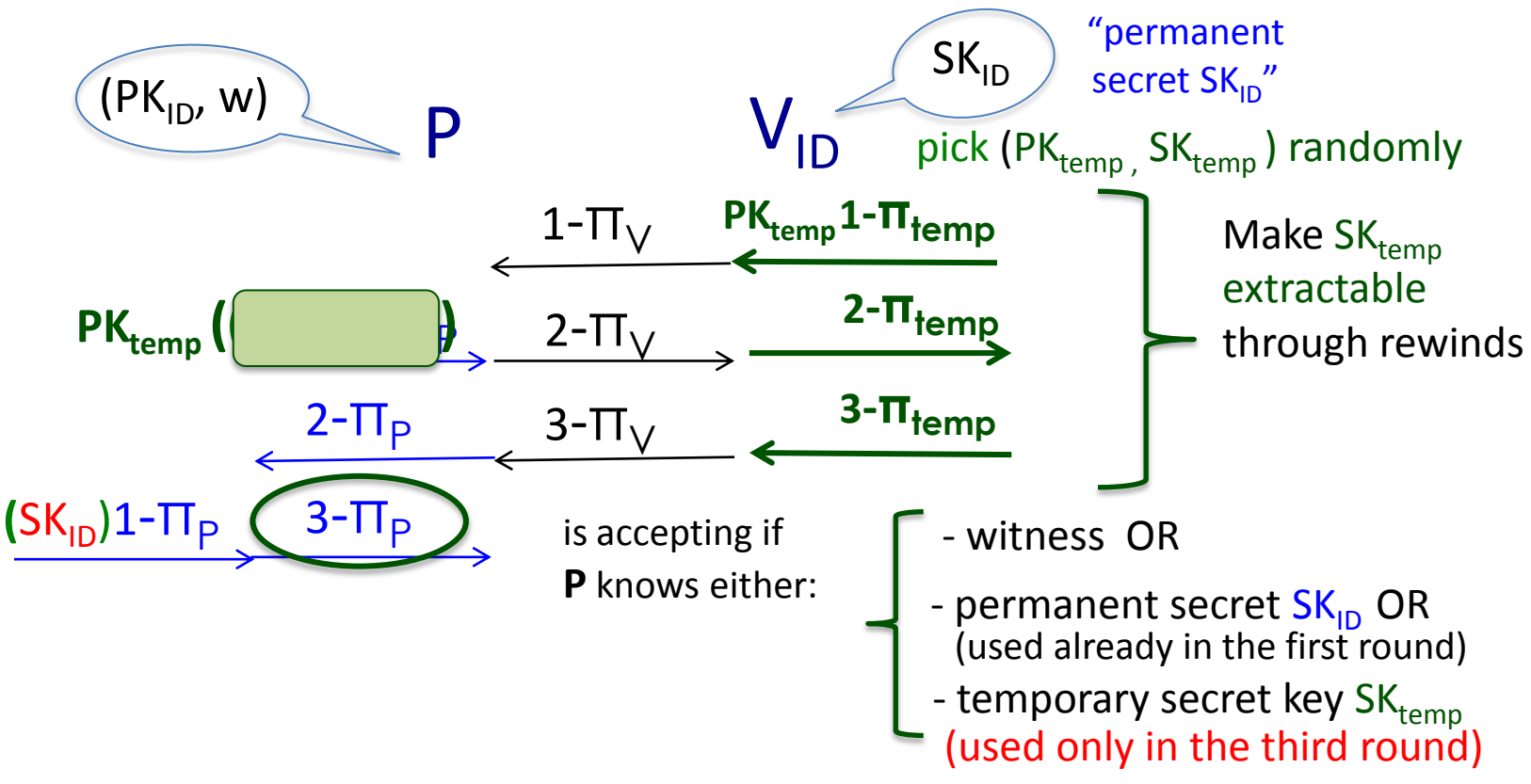
- In order to "solve" a session (played with a new identity) Sim has to change the view of the verifier (first play a bad msg, then a good msg)
- changing the view of V* skews the output distribution.

designing a successful simulation strategy seems problematic.

Outline

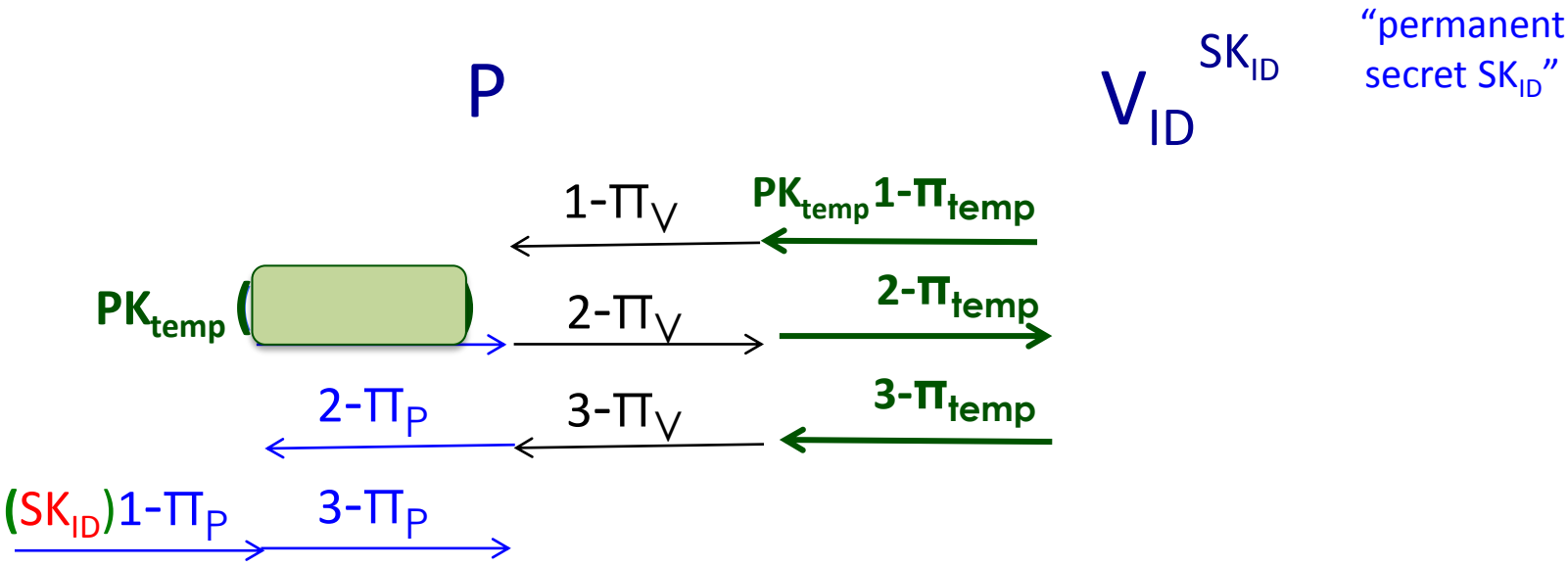
- Definitions
  - Concurrent Zero Knowledge
  - Bare Public Key (BPK) Model
  - Concurrent Zero Knowledge and Soundness in the BPK model
- Round-optimal Concurrent Zero Knowledge:
  - the issue of *all* zero-knowledge simulators
  - the difficulty of designing any alternative simulator
- Our technique

Our round-optimal concurrent ZK



KEY IDEA. Temporary secret key Sk_{temp} is used **only in the last msg** $3\text{-}\pi_P$.
(only *after the extraction*)

The simulator

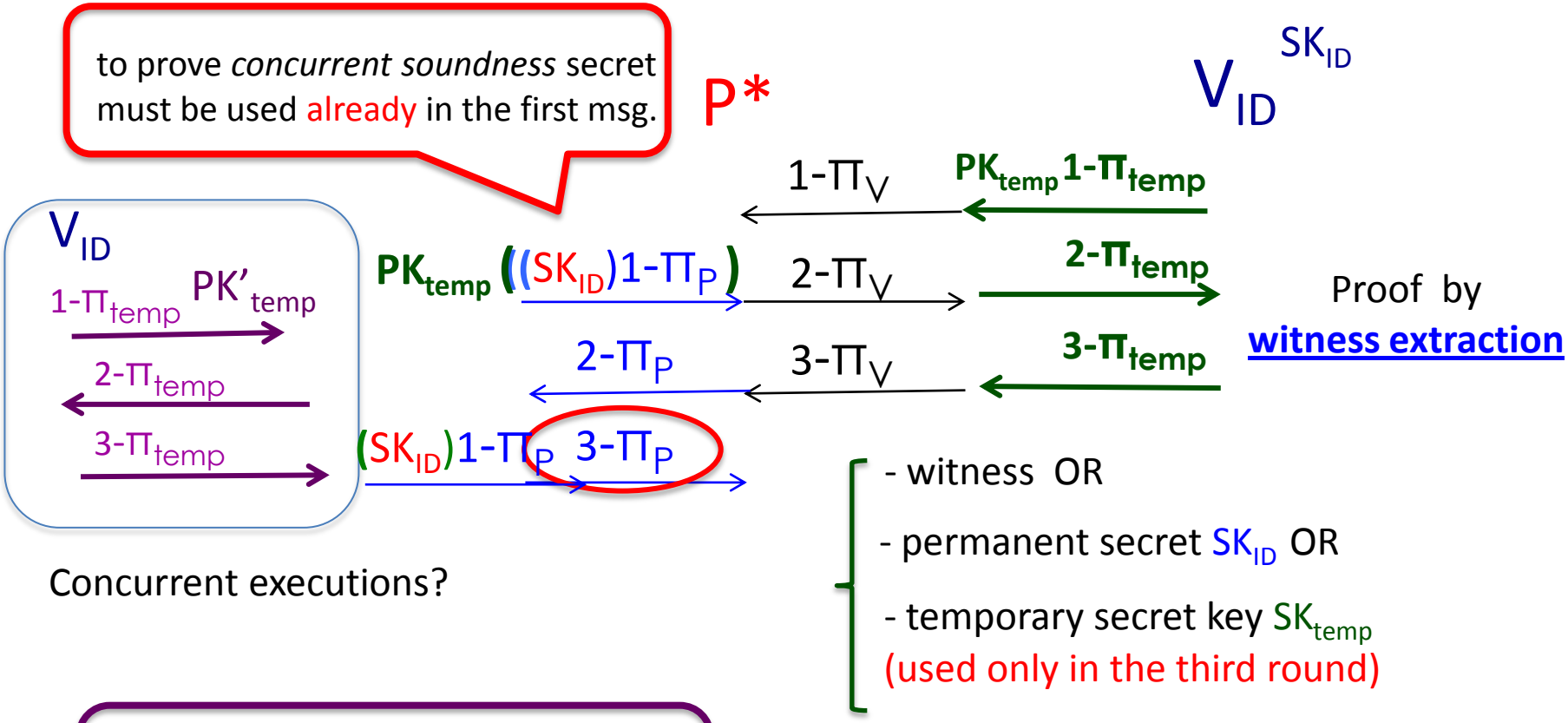


Two-mode simulation (allows to keep the main thread unchanged)

- to solve a session initiated by an unknown identity Sim extracts both permanent SK_{ID} and temporary key SK_{temp}, and computes the last msg using Sk_{temp} .
- to solve a session initiated by a known identity Sim runs in straight-line computing $3\text{-}\Pi_P$ using the permanent secret SK_{ID}.
- the view of V* in the two modes must be statistically indistinguishable.

Concurrent soundness?

to prove *concurrent soundness* secret must be used already in the first msg.



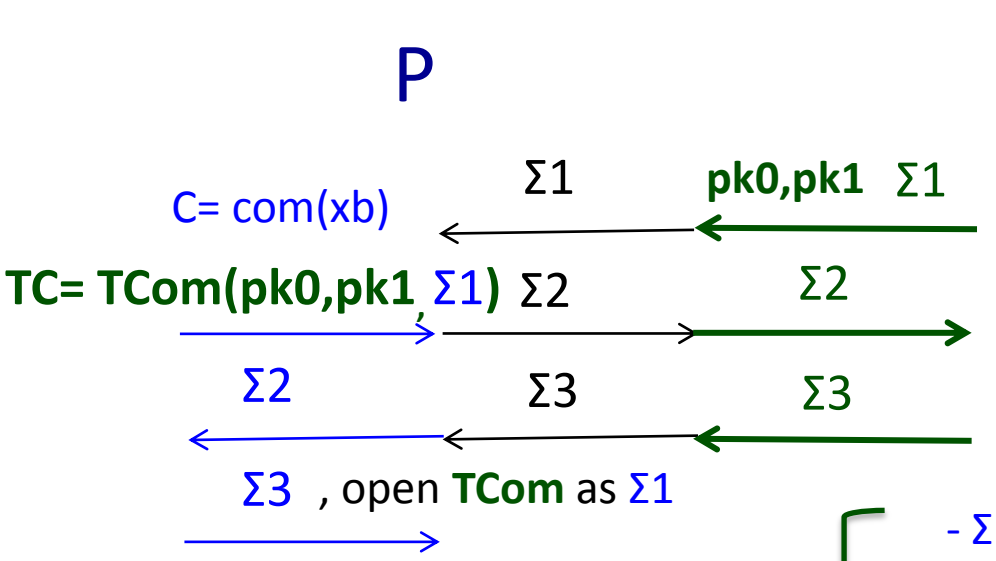
Proof by **witness extraction**

Concurrent executions?

- witness OR
- permanent secret SK_{ID} OR
- temporary secret key SK_{temp} (used only in the third round)

key point: the temporary keys used in concurrent sessions are *independent.*

Actual implementation



Pk_{temp} = pk0,pk1,
Sk_{temp} = trap0, trap1.

V_{ID} accepts if:

- Σ1 is the valid opening of **TC** AND (Σ1, Σ2, Σ3) is accepting.

(Σ1, Σ2, Σ3) is accepting iff:

- C is the commitment of xb **OR**
- P knows the witness

- Π_V Π_{temp} Π_P are implemented with Sigma Protocols.
- **TCom** is a two-round trapdoor commitment scheme.
- f is a OWP.

thanks