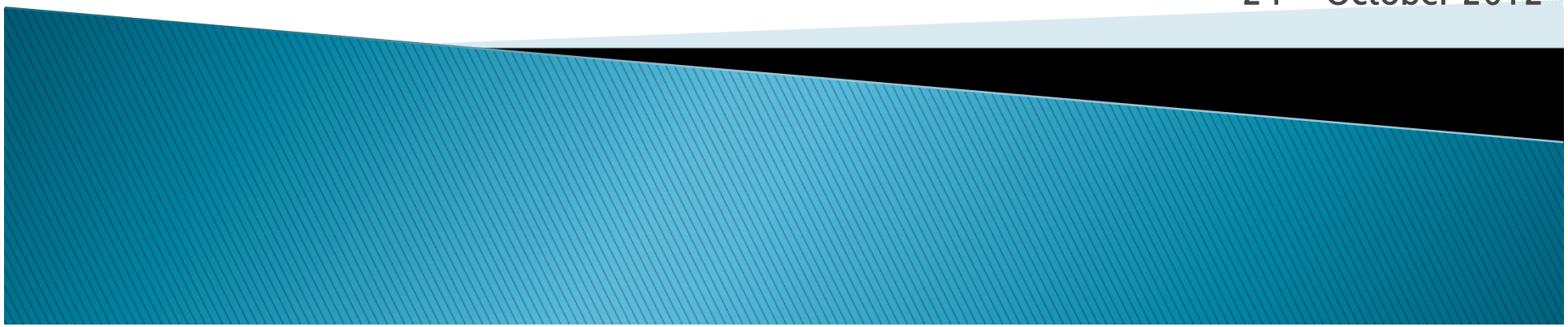
# Improving Protocol Validation by an IOSTS-based Passive Testing approach

## STV 2012 : 9th Workshop on SYSTEM TESTING AND VALIDATION

Pramila Mouttappa, Stephane Maag and Ana Cavalli
Institut Mines-Telecom/ Telecom SudParis
FRANCE

24th October 2012

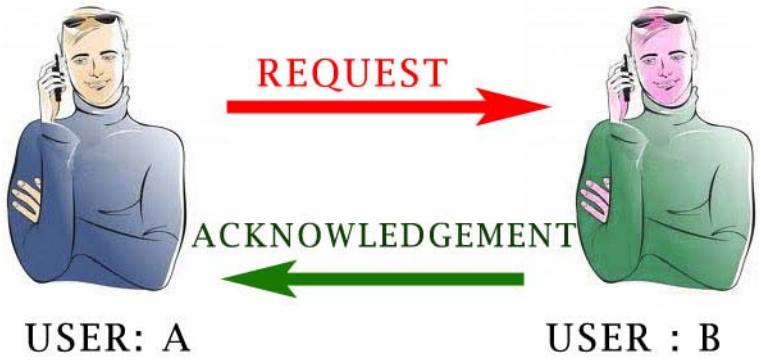


# Overview

- Problem Statement
- Motivation
- Our approach
- Registration Property and attack scenarios – SIP
  - Sequence Diagram
  - IOSTS model
  - SE model
- Parametric Trace Slicing
  - Example
- Evaluation – IOSTS property on the trace slices
- Final Evaluation – IOSTS property on the Implementation traces
- Prototype Output
- Prototype Tool
- Future Works

# Problem Statement

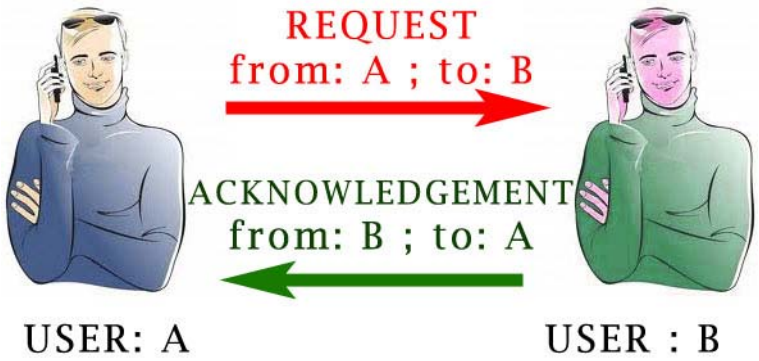
- Passive Testing
  - For checking both the data and control part of the protocol.
  - Enumeration of data values.
  - Specification dependent




(i) ONLY CONTROL PART

Invariant : Req / Ack
System Trace : *, Req,…,Ack, *,Req,*,Req,…
Verdict = **True**

# Problem Statement




(ii) CONTROL + DATA PART

Invariant : Req(A) / Ack(B)

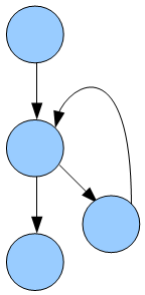
System Trace : *, Req(A),Ack(C), *, Req(A), Req(B),…

Verdict = **False or Inconclusive**

# Motivation

- IOSTS – Functional and Security attacks can be modeled.
- SYMBOLIC – Enumeration of data values is not required.
- REDUCES FALSE POSITIVE VERDICTS – Data relationship between messages.
- SYMBOLIC EXECUTION – Passive Testing.

# Our Approach



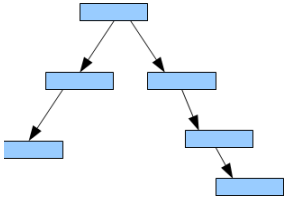
RFC / Specifications / User manual

Obtain behavior or attack scenarios to monitor

**IOSTS (M)** representation of behaviors/attack scenarios

Symbolic execution of IOSTS, SE(M)

Obtain the **Symbolic Execution Tree**

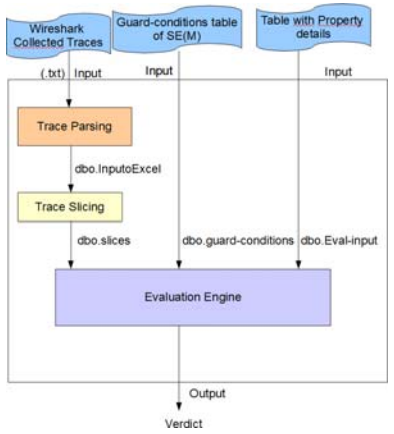


Extract the **Trace(SE(M)) and state details** (guard-conditions, Assignments)

**Traces from Wireshark** (.txt format)

Input

Input

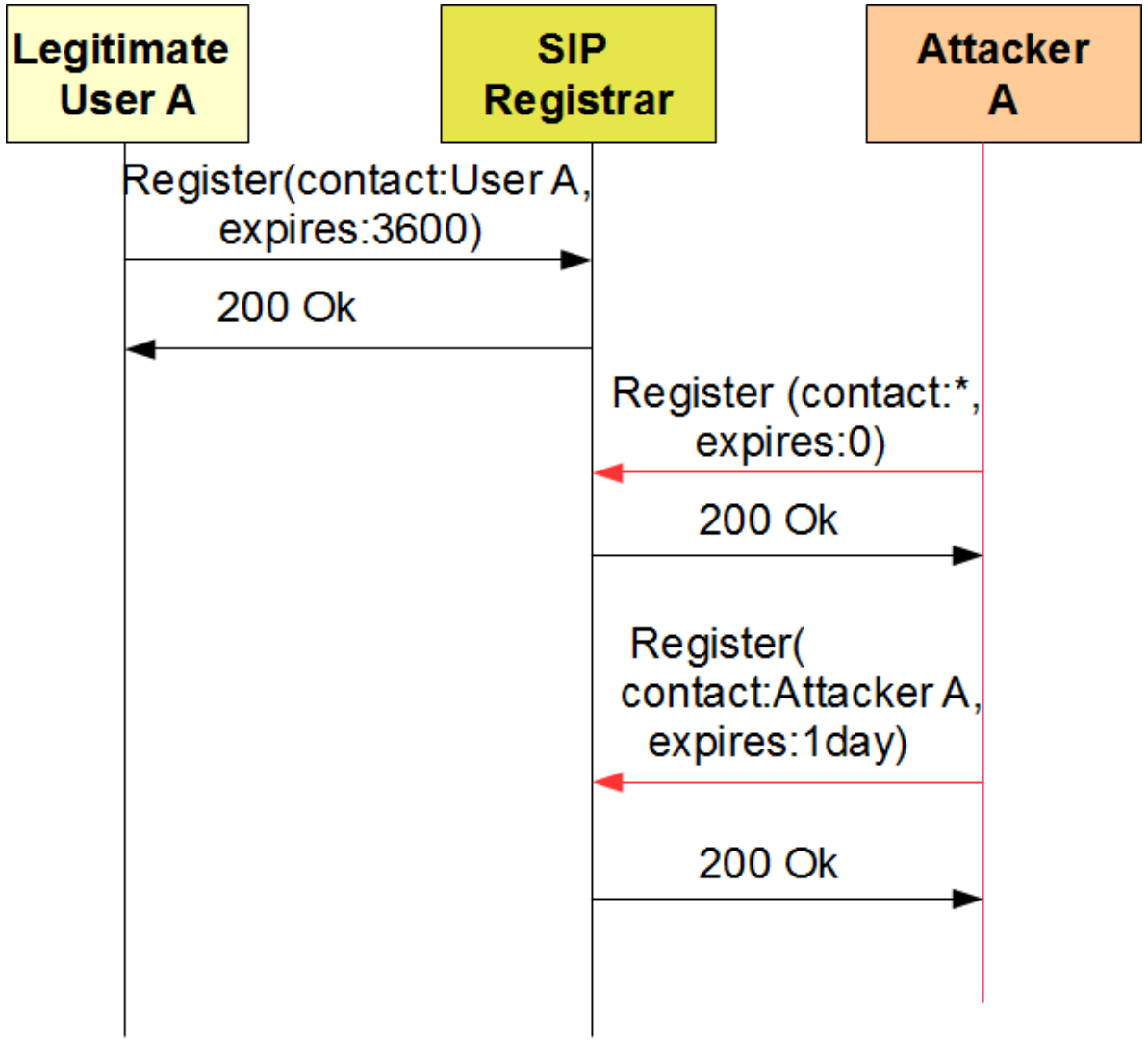



**Prototype Model** (**Trace Slicing & Evaluation algorithm implemented**)

Output

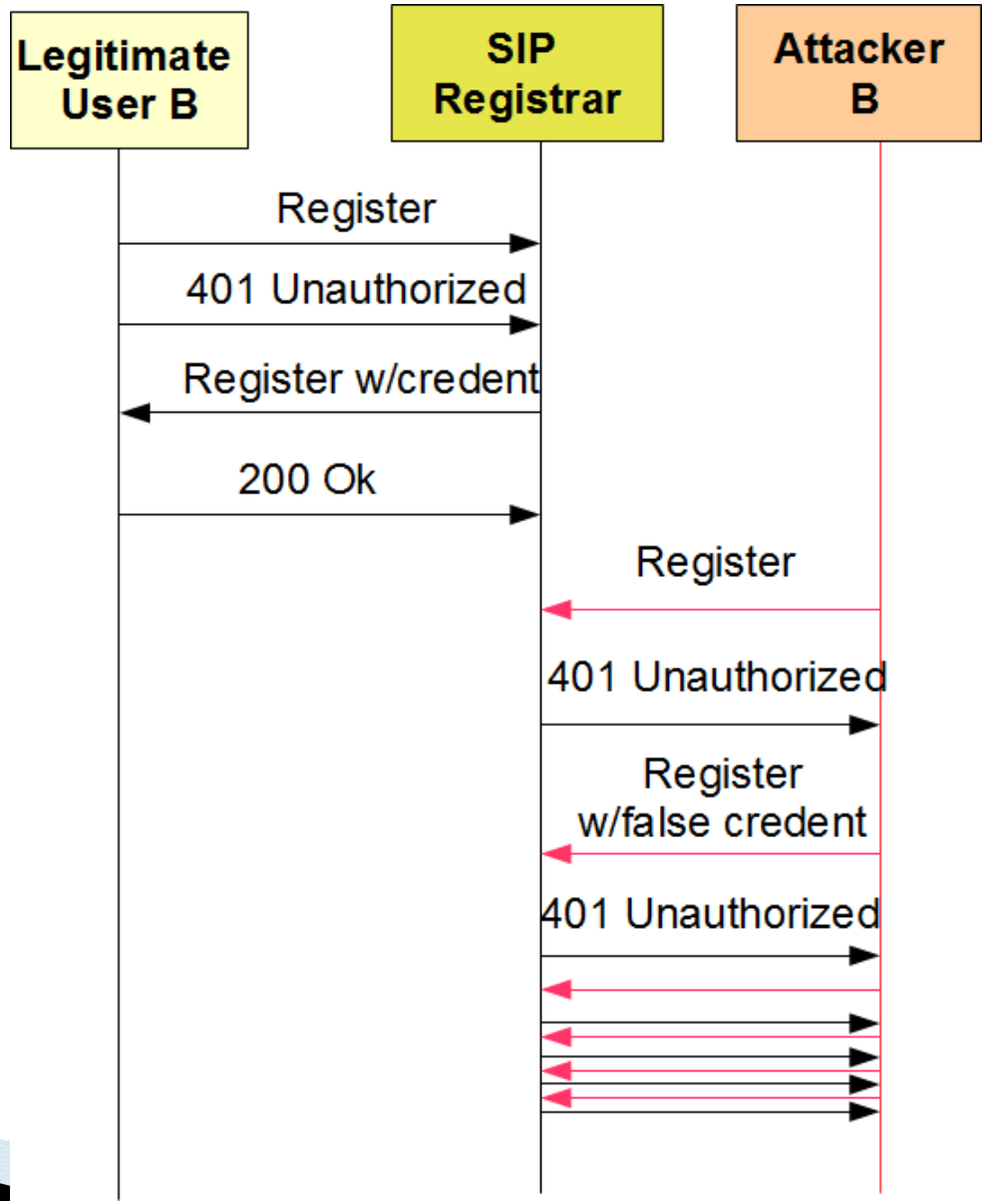
**Verdicts** (Pass/Attack-Pass/Fail/Inconclusive)

# Sequence Diagram – Registration Property and De-registration attack in SIP[1]

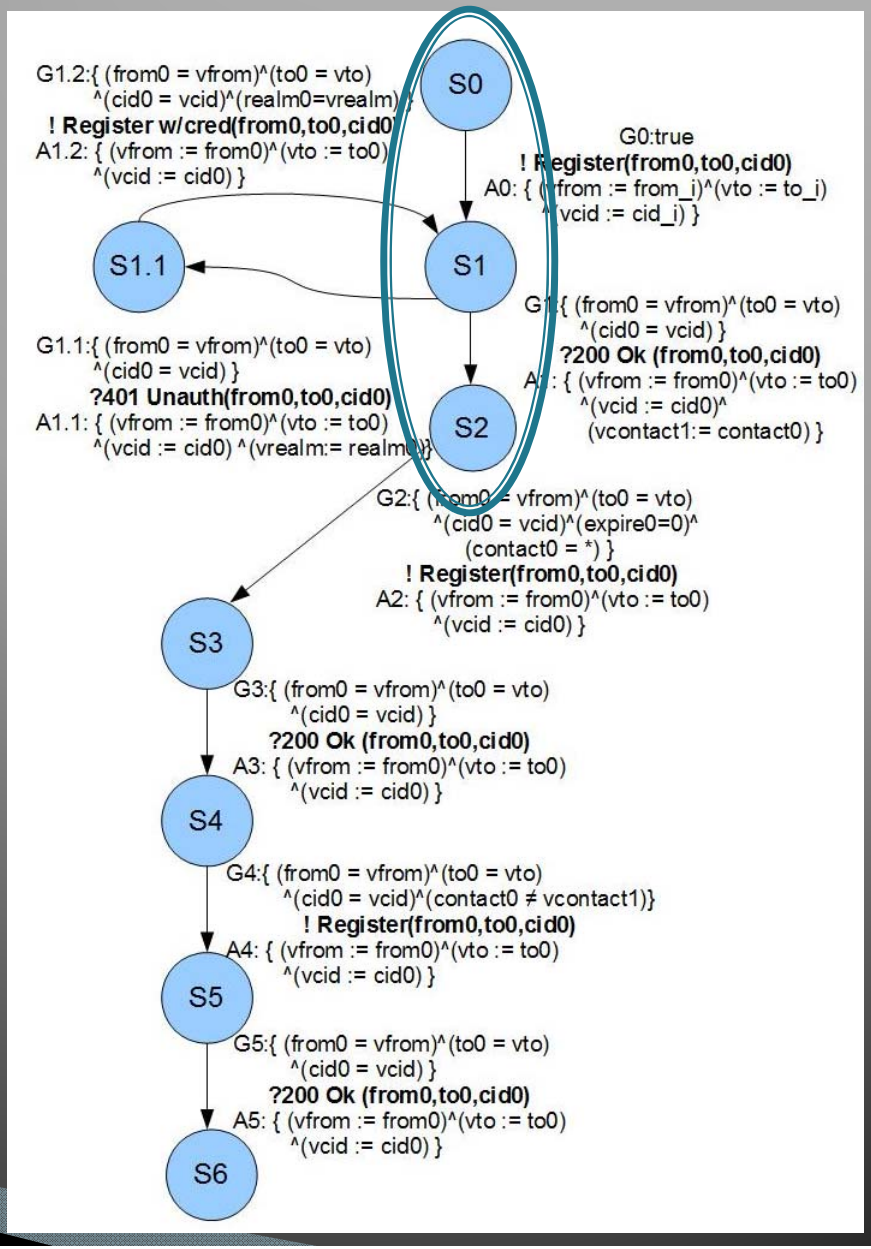



[1] J. Rosenberg , H. Schu... G. Camarillo, A. Johnston,J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol R... 2002.

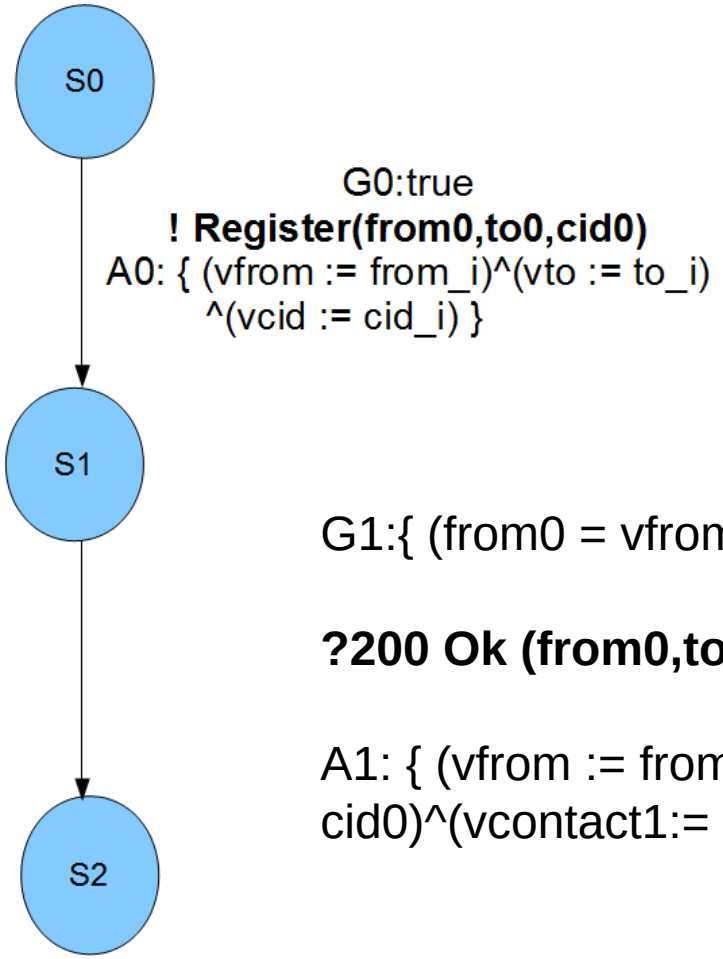
# Sequence Diagram – Registration Property and DoS attack in SIP[1]



# Registration Property and attack scenarios in SIP – IOSTS



# Registration Property and attack scenarios in SIP – IOSTS

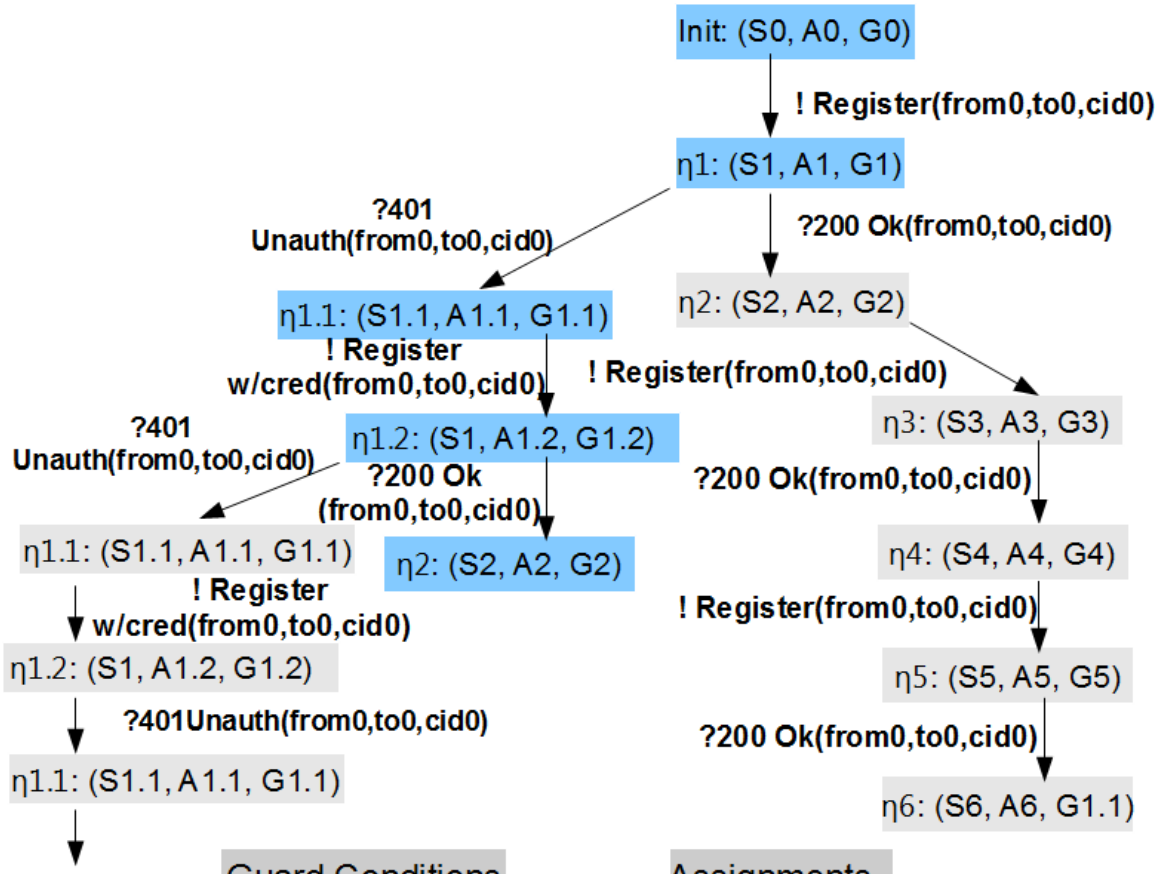



G1:{ (from0 = vfrom)^(to0 = vto) ^(cid0 = vcid) }

**?200 Ok (from0,to0,cid0)**

A1: { (vfrom := from0)^(vto := to0)^(vcid := cid0)^(vcontact1:= contact0) }

# Symbolic Execution of IOSTS



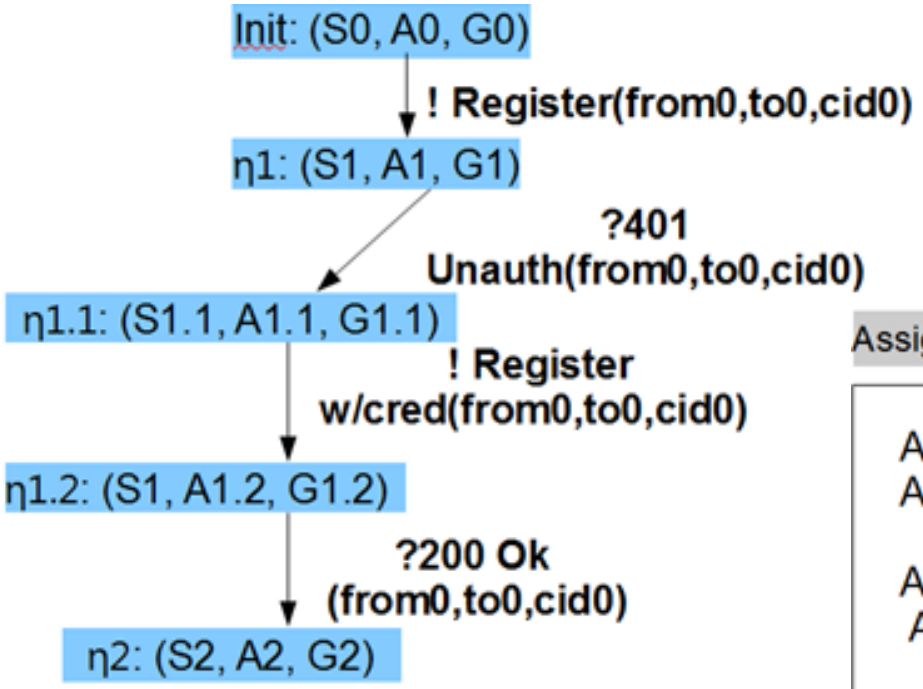

**Guard Conditions**

G0: true
G1: from0= vfrom^to0=vto^cid0=vcid
G2: from0= vfrom^to0=vto^cid0=vcid^ expire0=0^contact0 = *
G3: from0= vfrom^to0=vto^cid0=vcid
G4: from0= vfrom^to0=vto^cid0=vcid^ contact0 ≠ vcontact1
G5: from0= vfrom^to0=vto^cid0=vcid
G1.1:from0= vfrom^to0=vto^cid0=vcid
G1.2:from0= vfrom^to0=vto ^cid0 = vcid ^realm0 = vrealm

**Assignments**

A0: vfrom:= from_i, vto:= to_i, vcid:= cid_i
A1: vfrom:= from0, vto:= to0, vcid:= cid0, vcontact1:= contact0
A2: vfrom:= from0, vto:= to0, vcid:= cid0
A3: vfrom:= from0, vto:= to0, vcid:= cid0
A4: vfrom:= from0, vto:= to0, vcid:= cid0
A5: vfrom:= from0, vto:= to0, vcid:= cid0
A6: vfrom:= from0, vto:= to0, vcid:= cid0
A1.1: vfrom := from0,vto := to0,vcid := cid0, vcseq:= cseq0
A1.2: vfrom := from0, vto := to0, vcid := cid0

# Symbolic Execution of IOSTS




Guard Conditions

G0: true
G1: from0= vfrom^to0=vto^cid0=vcid
G2: from0= vfrom^to0=vto^cid0=vcid^
expire0=0^contact0 = *
G1.1:from0= vfrom^to0=vto^cid0=vcid
G1.2:from0= vfrom^to0=vto ^cid0 = vcid
^realm0 = vrealm

Assignments

A0: vfrom:= from_i, vto:= to_i, vcid:= cid_i
A1: vfrom:= from0, vto:= to0, vcid:= cid0,
vcontact1:= contact0
A2: vfrom:= from0, vto:= to0, vcid:= cid0
A1.1: vfrom := from0,vto := to0,vcid := cid0,
vcseq:= cseq0
A1.2: vfrom := from0, vto := to0, vcid := cid0

# Overview

- Problem Statement
- Motivation
- Define an IOSTS
- Registration Property and attack scenarios – SIP :
  - UML model
  - IOSTS model
  - SE model
- **Parametric Trace Slicing**
  - Example
- Evaluation – IOSTS property on the trace slices
- Final Evaluation – IOSTS property on the Implementation traces
- Prototype Output
- Prototype Tool
- Future Works

# Parametric Trace Slicing[2]

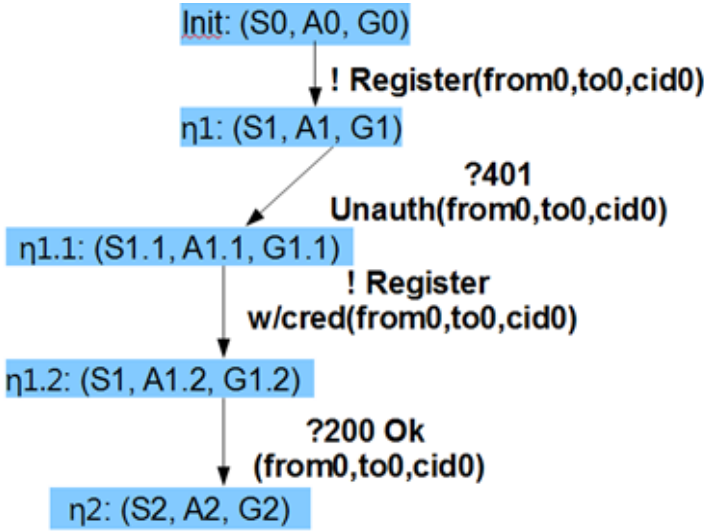
- Trace Monitoring technique
- Trace slices – Each parameter instance observed in the trace.
- The events corresponding to a particular parameter instance are grouped in the order they appear in the trace in a particular trace slice.
- Other events that are unrelated to the given parameter instance are dropped.

[2] F. Chen and G. Rosu, Parametric trace slicing and monitoring. TACAS'09.

# Parametric Trace Slicing – Example

- Consider a sample SIP trace,

! Register(From:Dav, To:Clara, CID:123)
?401 Unauth(From:Dav, To:Clara, CID:123)
! Register(From:Ann, To:Clara, CID:456)
! Register w/Credentials(From:Dav, To:Clara, CID
?200 Ok(From:Dav, To:Clara, CID:123)
?200 Ok(From:Ann, To:Clara, CID:456)




| Slices | Data Portion | Control Portion |
|---|---|---|
| Slice 1 | From:Dav, To:Clara, CID:123 | !Register?401Unauth!Register w/cred?200 Ok |
| Slice 2 | From:Ann, To:Clara, CID:456 | !Register?200Ok |

# Evaluation - IOSTS property on the trace slices

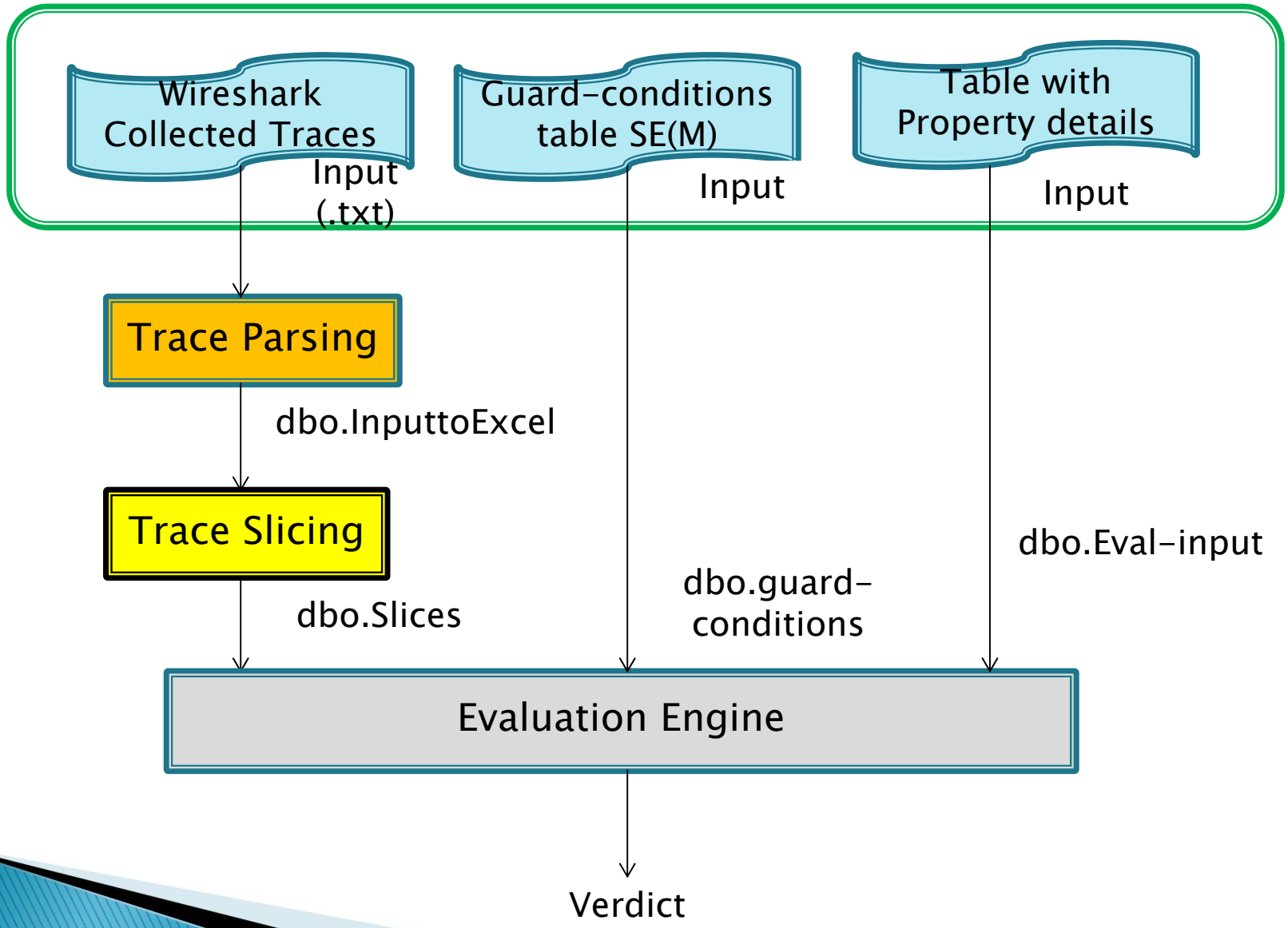
- Control portion : CP[Traces(SE(M)) ] \ Control portion(slice)
- Data portion:
  - Substitute concrete data values in place of symbolic values.
  - Satisfiability of each state in path of the SE – Guard condition check.
  - AttackSeq : 0 (Property Sequence), 1 (Attack Sequence)

| AttackSeq | Control Portion | Data Portion | Verdicts | Comments |
|---|---|---|---|---|
| 0 | Y | Y | Pass | Property satisfied. |
| 0 | Y | N | Fail | Property not satisfied. |
| 1 | Y | Y | Attack Pass | Attack sequence satisfied. |
| 0 or 1 | N | - | Inconclusive | Trace length insufficient to decide. |

# Final Evaluation – Property on the Implementation trace

- PASS, if (Every (Verdict(tsi) = Pass))
- Attack-PASS, if (Exists(Verdict(tsi) = Attack-Pass))
- FAIL, if [(Exists(Verdict(tsi) = Fail)) ^ (Exists(Verdict(tsi)≠ Attack-Pass))]
- INCONCLUSIVE, otherwise.

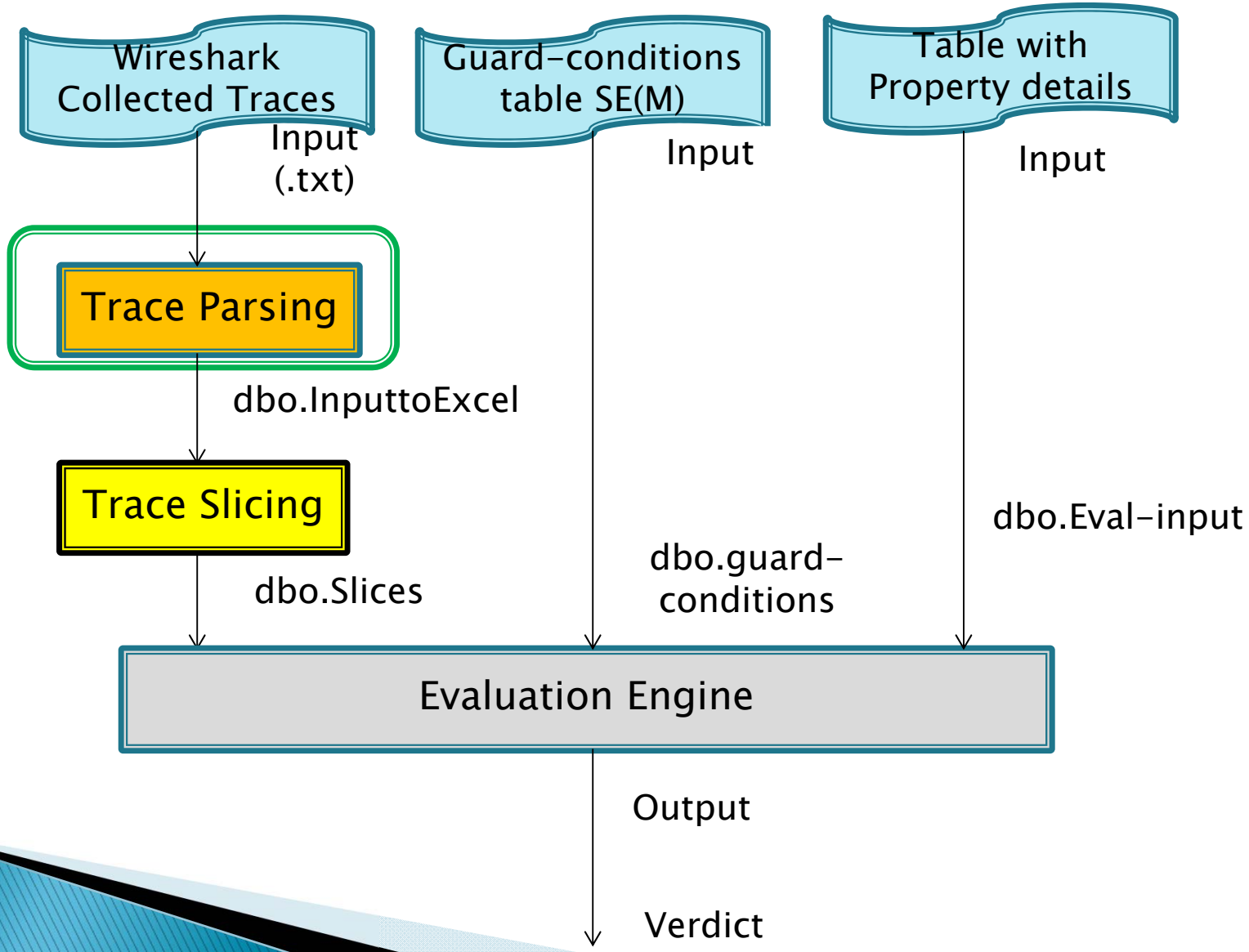
# Prototype Tool



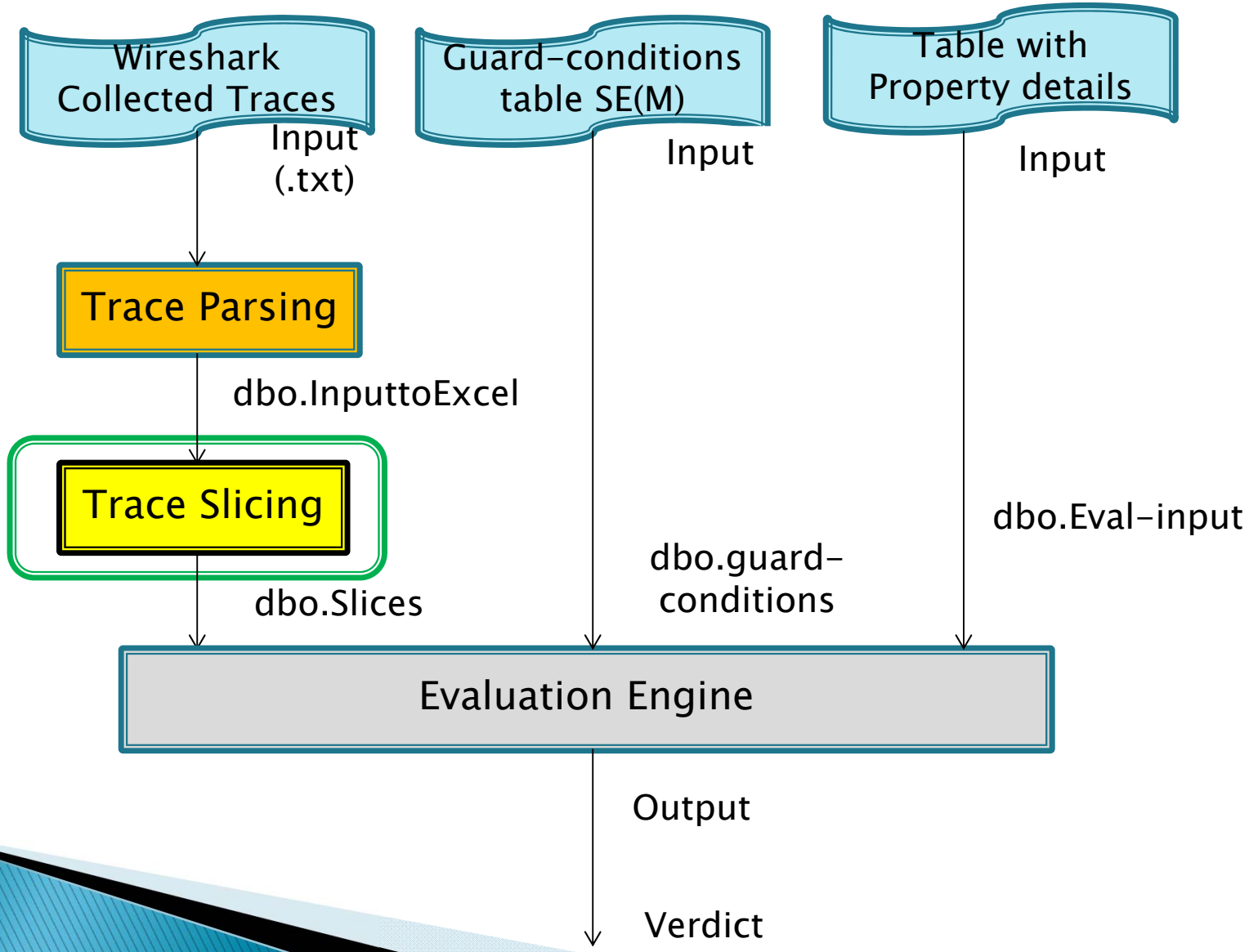
# Prototype – Inputs

- Captured trace using Wireshark (.txt format)
- Guard conditions – Symbolic state.
- Symbolic traces collected from the symbolic execution of the IOSTS property ((i.e.) the property that is to be verified on the IUT traces) + Symbolic state details.

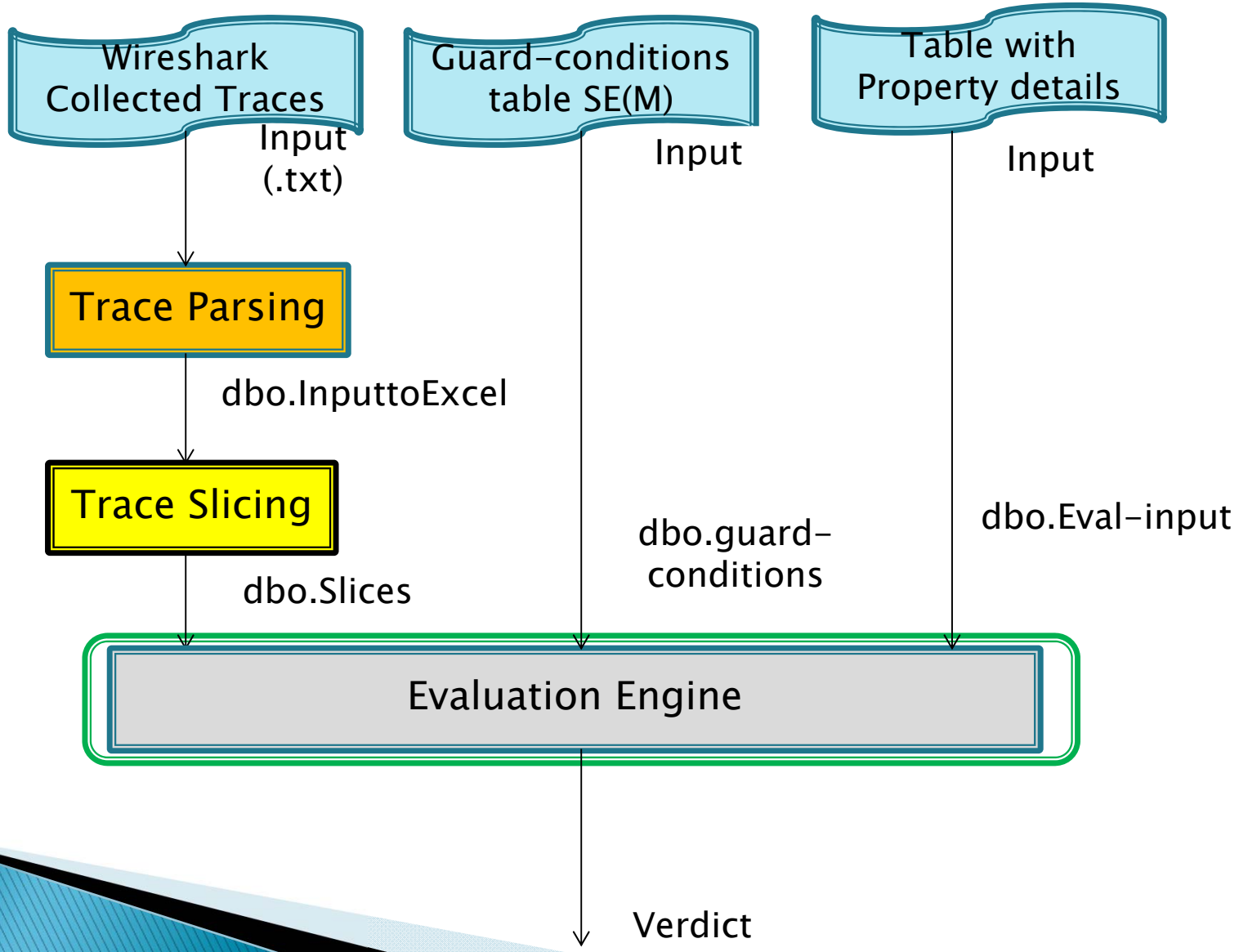
# Prototype Tool



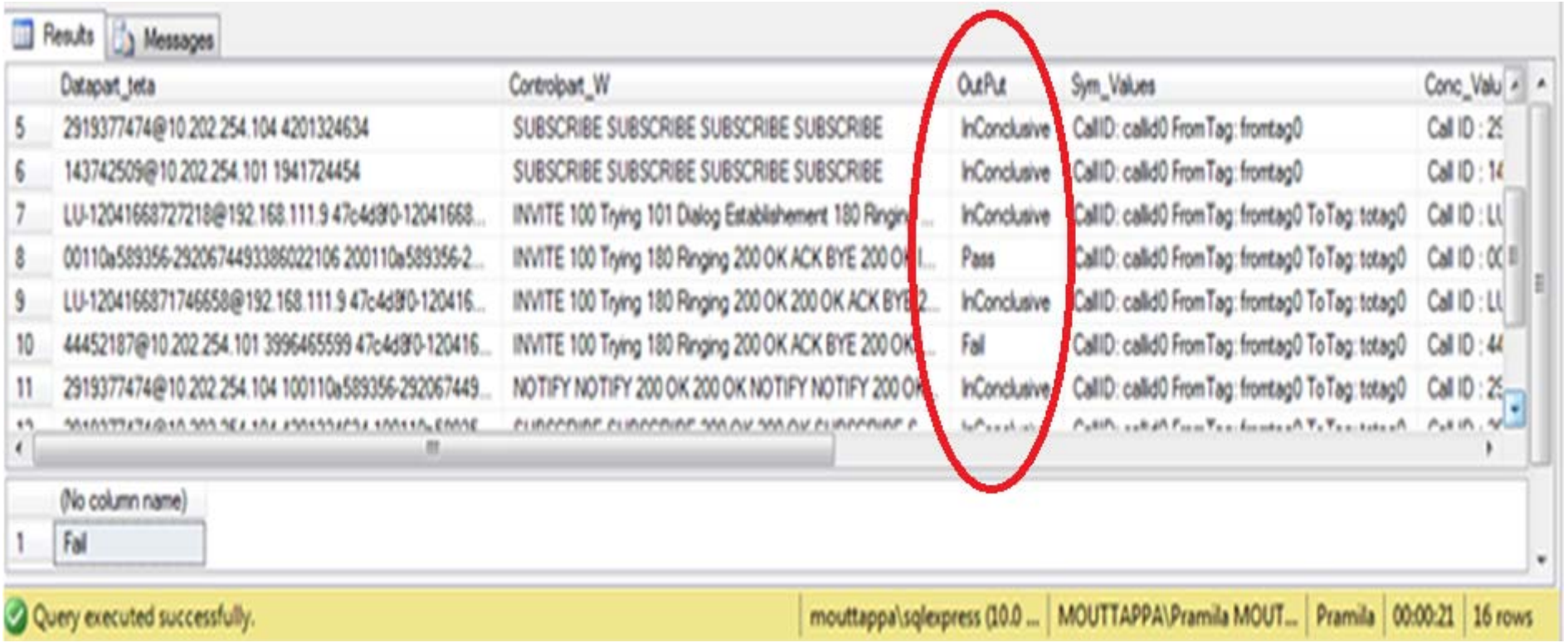
# Prototype Tool



# Prototype Tool



# Prototype – Evaluation Output



Results | Messages

| | Datapart_teta | Controlpart_W | OutPut | Sym_Values | Conc_Valu |
|---|---|---|---|---|---|
| 5 | 2919377474@10.202.254.104 4201324634 | SUBSCRIBE SUBSCRIBE SUBSCRIBE SUBSCRIBE | InConclusive | CallID: callid0 FromTag: fromtag0 | Call ID : 2 |
| 6 | 143742509@10.202.254.101 1941724454 | SUBSCRIBE SUBSCRIBE SUBSCRIBE SUBSCRIBE | InConclusive | CallID: callid0 FromTag: fromtag0 | Call ID : 1 |
| 7 | LU-12041668727218@192.168.111.9 47c4d8f0-12041668... | INVITE 100 Trying 101 Dialog Establishement 180 Ringi... | InConclusive | CallID: callid0 FromTag: fromtag0 ToTag: totag0 | Call ID : L |
| 8 | 00110a589356-292067449338602210 6 200110a589356-2... | INVITE 100 Trying 180 Ringing 200 OK ACK BYE 200 O... | Pass | CallID: callid0 FromTag: fromtag0 ToTag: totag0 | Call ID : 0 |
| 9 | LU-120416687174665 8@192.168.111.9 47c4d8f0-120416... | INVITE 100 Trying 180 Ringing 200 OK 200 OK ACK BY... | InConclusive | CallID: callid0 FromTag: fromtag0 ToTag: totag0 | Call ID : L |
| 10 | 44452187@10.202.254.101 3996465599 47c4d8f0-120416... | INVITE 100 Trying 180 Ringing 200 OK ACK BYE 200 O... | Fail | CallID: callid0 FromTag: fromtag0 ToTag: totag0 | Call ID : 4 |
| 11 | 2919377474@10.202.254.104 100110a589356-292067449... | NOTIFY NOTIFY 200 OK 200 OK NOTIFY NOTIFY 200 O... | InConclusive | CallID: callid0 FromTag: fromtag0 ToTag: totag0 | Call ID : 2 |

| | (No column name) |
|---|---|
| 1 | Fail |

Query executed successfully. | mouttappa\sqlexpress (10.0 ... | MOUTTAPPA\Pramila MOUT... | Pramila | 00:00:21 | 16 rows

# Experimental Results

| Trace | No. Packets | No. Slices | Trace Output without errors | | | | Trace Output with errors and attacks | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | P | F | I | Final O/P | P | F | I | AP | Final O/P |
| 1 | 122 | 58 | 24 | – | 34 | I | 22 | 2 | 34 | – | F |
| 2 | 199 | 85 | 12 | – | 73 | I | 10 | 2 | 72 | 1 | AP |
| 3 | 280 | 280 | 140 | – | 140 | I | 138 | 2 | 140 | – | F |
| 4 | 4154 | 4146 | 2073 | – | 2073 | I | 2070 | 1 | 2073 | 2 | AP |
| 5 | 10000 | 5458 | 1365 | – | 4073 | I | 1365 | – | 4092 | 1 | AP |

# Future Works

- Reduce Number Of Inconclusive verdicts
  - Filtering – Reduce number of Slices.
  - Time constraints.

# QUESTIONS ?