



# Analysis of optimistic multi-party contract signing

Rohit Chadha<sup>1,2</sup>, Steve Kremer<sup>3</sup>, Andre Scedrov<sup>1</sup>

<sup>1</sup>University of Pennsylvania

<sup>2</sup>University of Sussex

<sup>3</sup>Université Libre de Bruxelles



# Digital Contract signing

- Use digital signatures to sign a contract over a network
- Special instance of fair exchange protocols
- Important issue for secure electronic commerce
- Naive 2-party example :

$A \rightarrow B : \text{Sig}_A (\text{contract})$

$B \rightarrow A : \text{Sig}_B (\text{contract})$



# Digital Contract signing

- Use digital signatures to sign a contract over a network
- Special instance of fair exchange protocols
- Important issue for secure electronic commerce
- Naive 2-party example :

$A \rightarrow B : \text{Sig}_A (\text{contract})$

$B \rightarrow A : \times$

- Bob may be malicious and not send his signature
- **Asymmetry** : someone must be the first to send his signature



# Properties of Contract Signing

- Fairness
  - If A can get B's signature, then B can get A's signature and vice-versa
- Timeliness
  - Avoids that a participant gets stuck
- Advantage
  - A participant has an advantage if
    - he has a strategy to complete the exchange
    - and he has a strategy to abort the exchange
- Abuse-freeness (provable advantage)
  - Avoids that a participant can prove to an external party that he has the power to choose the outcome of the protocol



# Evolution of contract signing

In 1980, Even & Yacobi showed that **no fair deterministic contract signing protocol exists without the participation of a trusted party.**

- **Randomized protocols**
- **Trusted Party intervenes**
  - Use trusted party as a delivery authority
  - May cause a bottleneck ...
- **Trusted Party intervenes only in case of problem (optimistic approach)**
  - More complex, and more error-prone ...

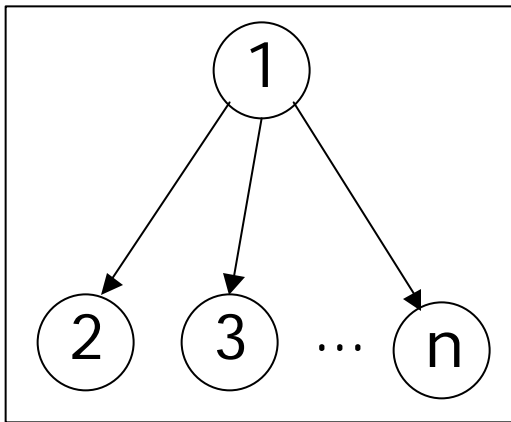


# Formal methods & contract signing

- [Shmatikov, Mitchell, 2000]
    - Model-checker Murphi
    - invariant checking
  - [Chadha, Kanovich, Scedrov, 2001]
    - Specification in MSR
    - inductive proofs
  - [Kremer, Raskin, 2002]
    - Model-checker Mocha
    - ATL (temporal logic with game semantics)
  - [Chadha, Mitchell, Scedrov, Shmatikov 2003]
    - general results (protocol independent) on advantage
- ⇒ Only 2-party contract signing protocols have been studied

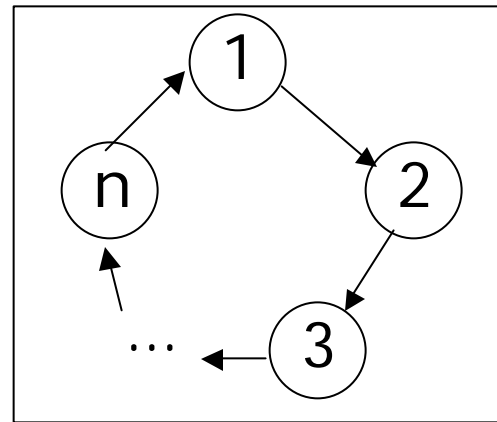
# Topologies

- Unlike for 2-party protocols, the different instances of fair exchange protocols differ significantly in the multi-party case



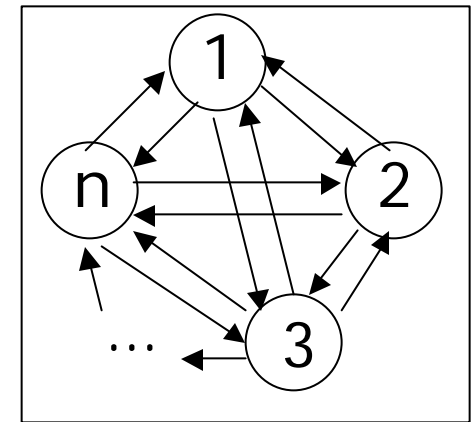
1-to-many

non-repudiation and  
certified e-mail



ring topology

barter



full graph

contract signing

- Contract signing requires the most complicated protocols



# Multi-party contract signing

- $n$  participants want to sign a contract
- Properties for a honest participant must hold against **any coalition of dishonest participants**, *i.e.*, against up to  $n-1$  dishonest participants
- Every participant must receive the signature of **all** other participants (topology is a **full graph**)



# Multi-party protocols

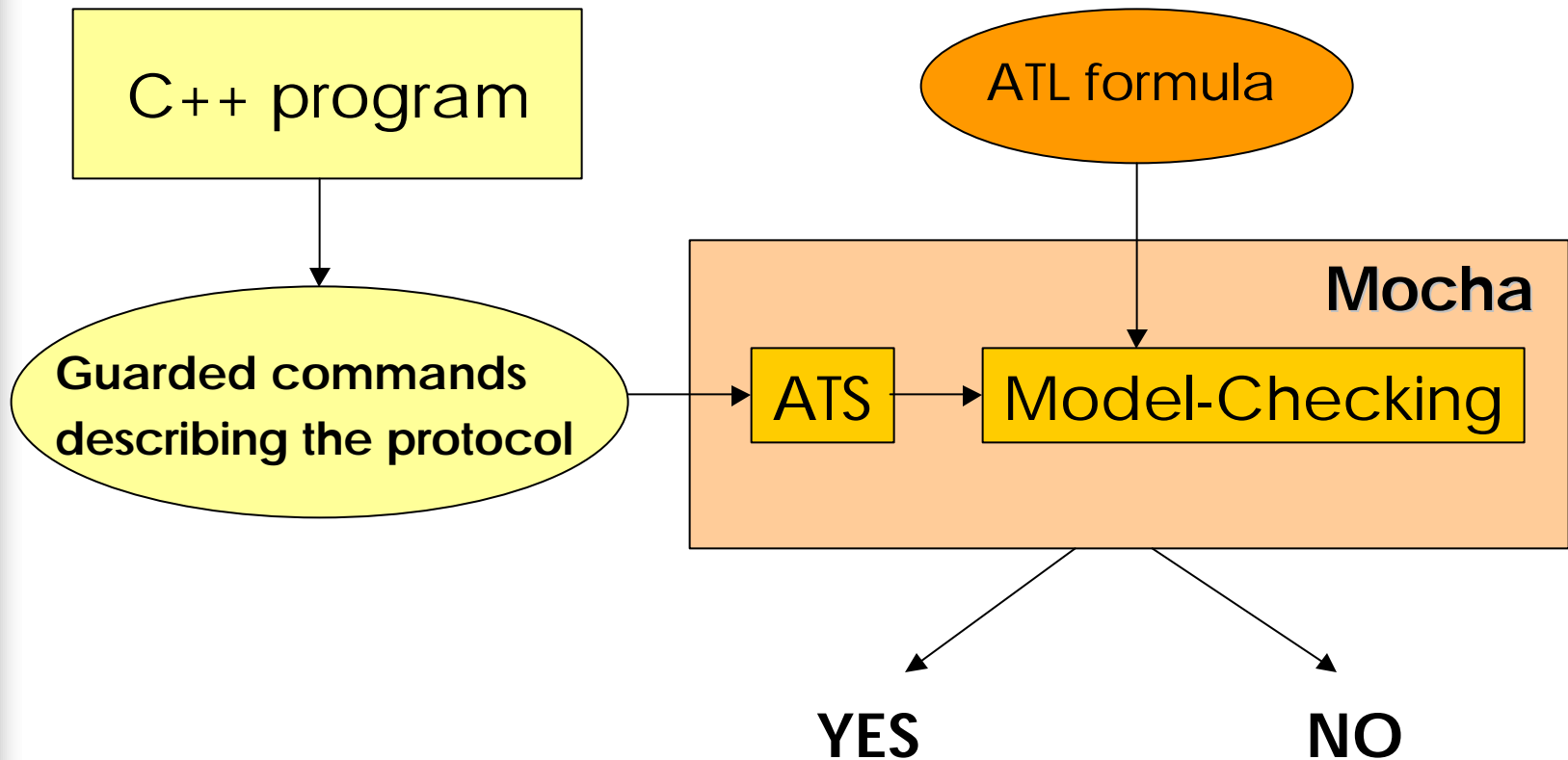
- Astonishingly few so far
- [Asokan, Baum-Waidner, Schunter, Waidner, T.R. 1998]  
Optimistic synchronous multi-party contract signing
- [Baum-Waidner, Waidner, T.R. 1998 & ICALP 2000]  
Optimistic asynchronous multi-party contract signing
- [Garay, MacKenzie, DISC 1999]  
Optimistic asynchronous multi-party contract signing
- [Baum-Waidner, Waidner, ICALP 2001]  
Optimistic asynchronous multi-party contract signing with reduced number of rounds



# Protocol model

- All participants are **players**
- **2 versions** of each player described using guarded commands
  - **honest** : follow the protocol
  - **dishonest** : may send messages out of order and continue the main protocol after contacting the trusted party
- Messages are **immediately available for reading**
- Only structural flaws are considered
  - no modelling of the cryptographic primitives
- Mocha cannot handle parametric specifications
  - Small **C++ programs** for the GM protocol and the BW protocol, that generate the Mocha specification for a given number of participants

# The model-checker Mocha





# The BW protocol [Baum, Waidner, ICALP 2000]

- Rather simple protocol, with symmetric behaviour of each participant
- T can overturn aborts
- We used Mocha to verify fairness for  $n=2, \dots, 5$ , but no flaw was found
- The basic protocol does not aim to provide abuse-freeness
- Non-standard definition of contract
  - a special protocol for verifying the validity of a contract is defined



# GM protocol [Garay, MacKenzie, DISC 1999]

- **Recursive** description of the protocol
- The protocol is divided into  $n$  levels
  - In each protocol level specific promises are used
  - Promises are implemented using **private contract signatures** (convertible designated verifier signatures)
- The  $i$ -level protocol is triggered when  $P_i$  receives  $i$ -level promises from  $P_{i+1}$  through  $P_n$
- In  $i$ -level protocol participants  $P_i$  through  $P_1$  exchange  $i$ -level promises
  - They agree on the contract with promises (not signatures)
- $P_i$  through  $P_1$  close higher level protocols
- After the  $n$ -level protocol actual signatures are exchanged



# GM main protocol for $P_i$

$P_i$

$P_{i-1}$

...

$P_1$

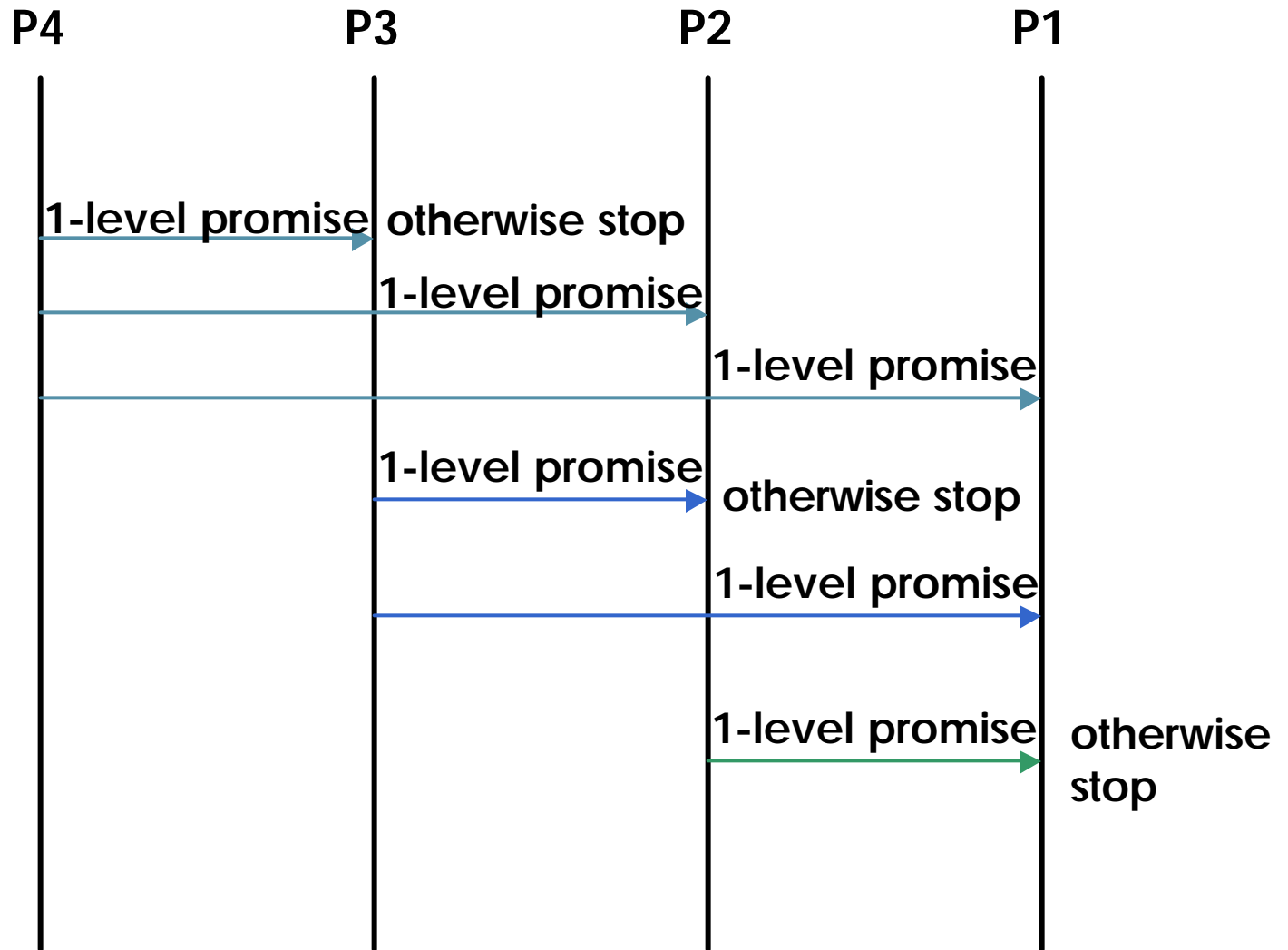
Distribute 1-level promises

(i-1) level protocol

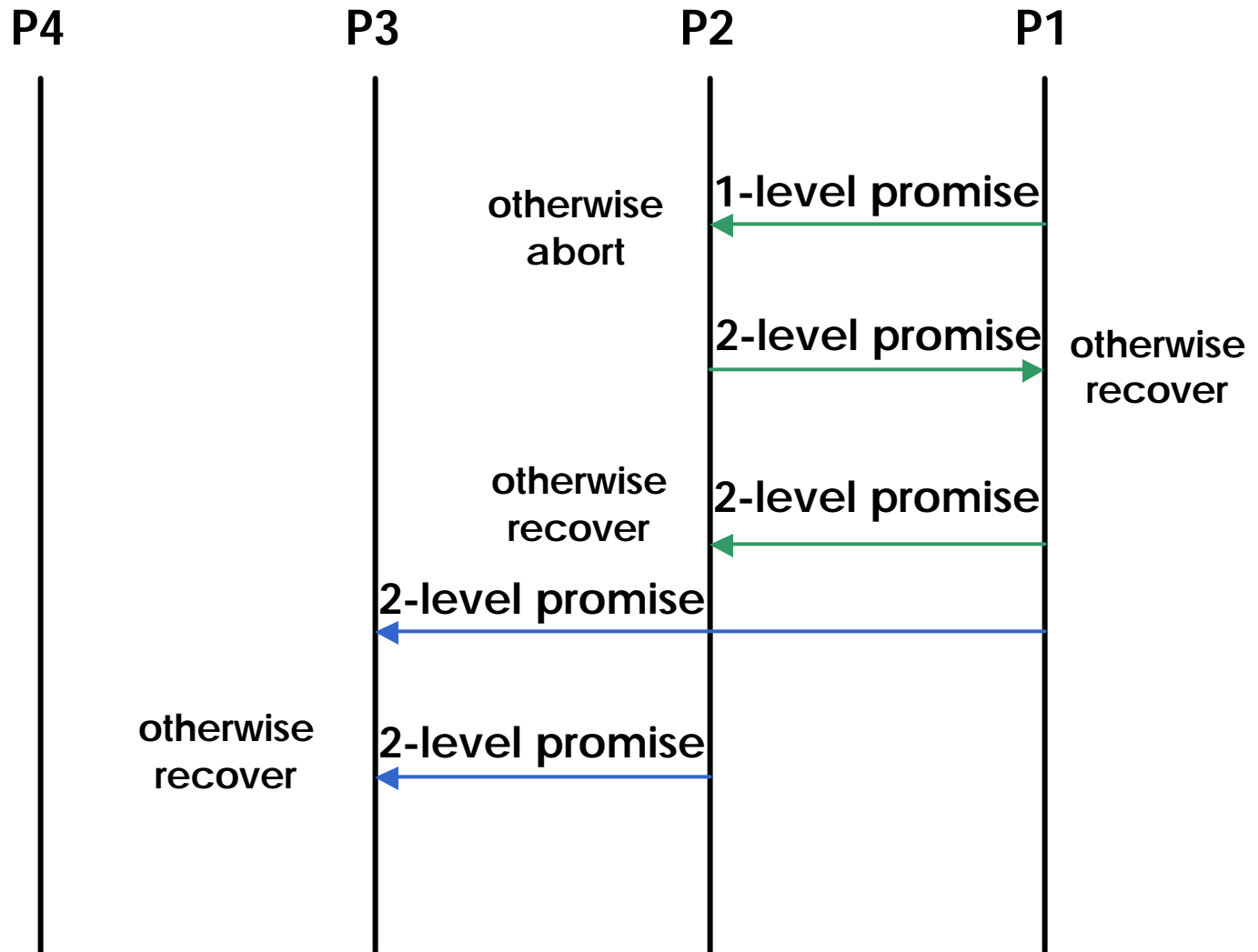
Collect (i-1) level promises

Exchange i-level promises

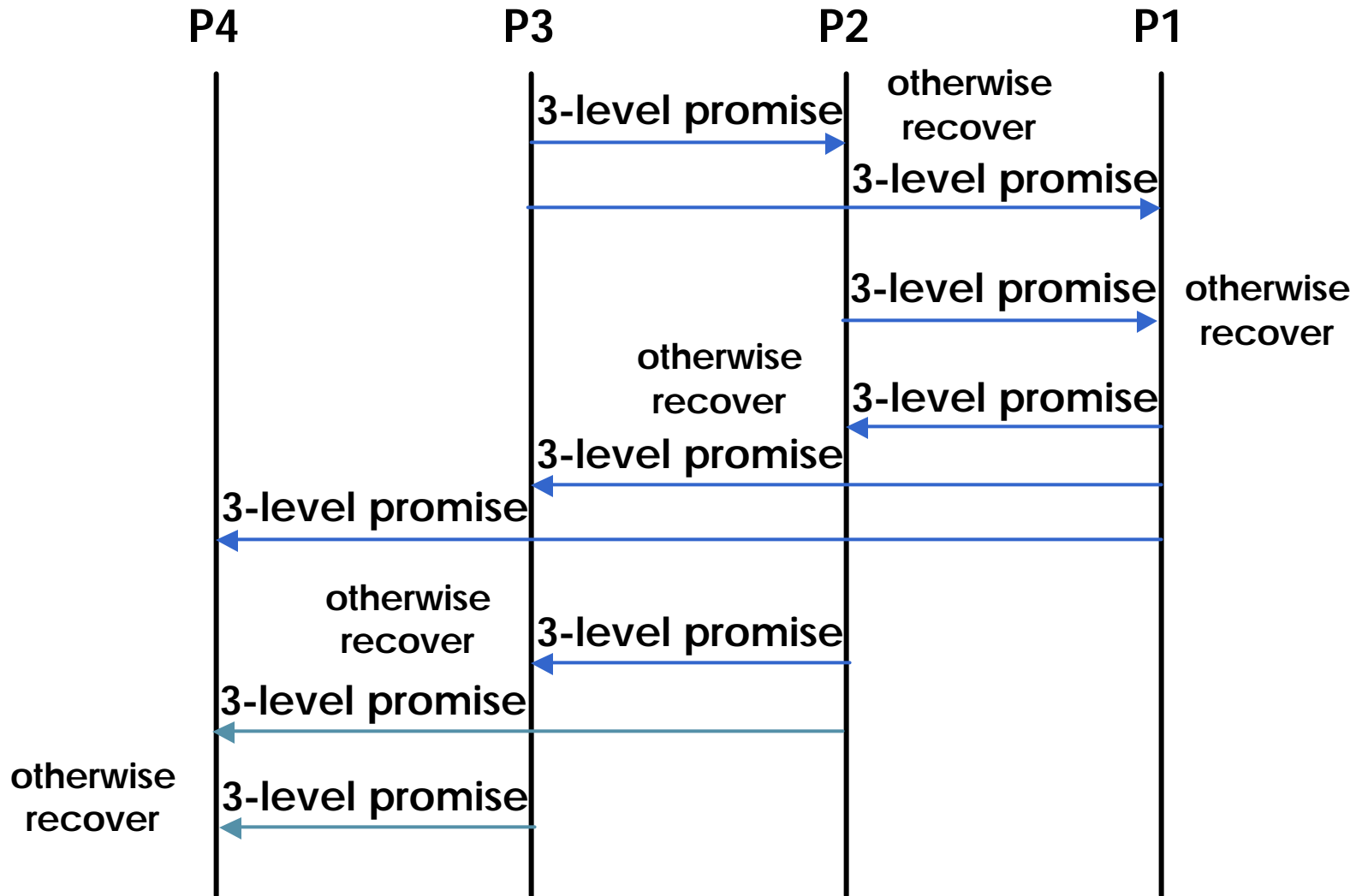
# GM main prot. (4 participants)



# GM main prot. (4 participants)



# GM main prot. (4 participants)









# GM abort and resolve for $P_i$

- To abort,  $P_i$  sends to T

$S_{P_i}(m, P_i, (P_1, \dots, P_n), \text{abort})$

- To resolve,  $P_i$  sends to T

$S_{P_i}(\{PCS_{P_j}(m, k_j), P_i, T\} (j \in \{1 \dots n\} \setminus \{i\}), S_{P_i}(m, 1))$

where

- if  $j > i$ ,  $k_j$  is the maximum level of a promise received from  $P_j$  on  $m$
- if  $j < i$ ,  $k_j$  is the maximum level of promises received from each of the participants  $P_{j'}$ , with  $j' < i$



# GM protocol for T

- Each participant may contact T **only once**
- T replies with a **resolved contract** or an **abort** token
- T may **overturn** an abort, but never a resolve
- T maintains the following information for each contract to decide when to overturn an abort
  - validated: a boolean indicating whether the contract has been validated or not
  - S: the set of indices of parties that have aborted
  - F: set of indices of parties which help T to decide when to overturn an abort



# An attack on abuse-freeness

- Note that  $P_1$  cannot abort
- Abort responses include the participants that have aborted
- If  $P_1$  receives an abort from T he must have send a resolve request
- Use T as an oracle :
  - When T receives a resolve request T verifies all promises and, by answering to  $P_1$ , provides evidence that all participants have started the protocol



## An attack on abuse-freeness (2)

- Consider the protocol instance where  $n=3$
- Using Mocha, we show that abuse-freeness does not hold for a honest  $P_3$   
 *$P_1$  and  $P_2$  have a strategy to reach a state where*
  - *$P_1$  has an abort reply and*
  - *$P_1$  and  $P_2$  have a strategy to obtain  $P_3$ 's signature*
  - *honest  $P_3$  does not have a strategy to obtain  $P_1$ 's and  $P_2$ 's signature*



## An attack on abuse-freeness (3)

- At the beginning P2 aborts
- P1 tries to resolve, but gets an abort reply from T, which he can show to Charlie
- At that point P1 and P2 can choose the outcome
  - stop the protocol : P3 is not able to overturn the abort
  - complete the protocol in an optimistic way
- Easy fix: make abort replies to different participants indistinguishable



# An attack on fairness

- The first attack was discovered when noticing an error in the proof
- Consider the protocol instance where  $n=4$
- Using Mocha, we show that fairness does not hold for a honest  $P_2$

*There exists a path such that*

- *$P_1, P_3$  and  $P_4$  have  $P_2$ 's signature*
  - *there exists a path such that  $P_2$  does not obtain all other signatures*
- Similar attacks can be shown against  $P_1$  and  $P_3$
  - Using Mocha we did not discover any attack on fairness holds for  $n=3$



## An attack on fairness (2)

- $P_1$ ,  $P_3$  and  $P_4$  collude against  $P_2$
- $P_3$  aborts at the beginning
  - T adds  $P_3$  to S
- $P_1$  resolves, but T responds with an abort
  - T adds  $P_1$  to S and  $P_2$  to F
- $P_2$  tries to recover, but as  $P_2$  is in F, T responds with an abort
- $P_4$  resolves and T overturns the abort



## An attack on fairness (3)

- More generally the attack scenarios are as follows
  - dishonest  $P_{k1}$  aborts but continues the protocol
  - dishonest  $P_{k2}$  tries to recover but does not succeed
    - as a side-effect he adds one or several participants to the set  $F$
  - honest  $P_{k3}$  tries to recover but does not succeed
  - dishonest  $P_{k4}$  recovers and overturns the abort



# Conclusion

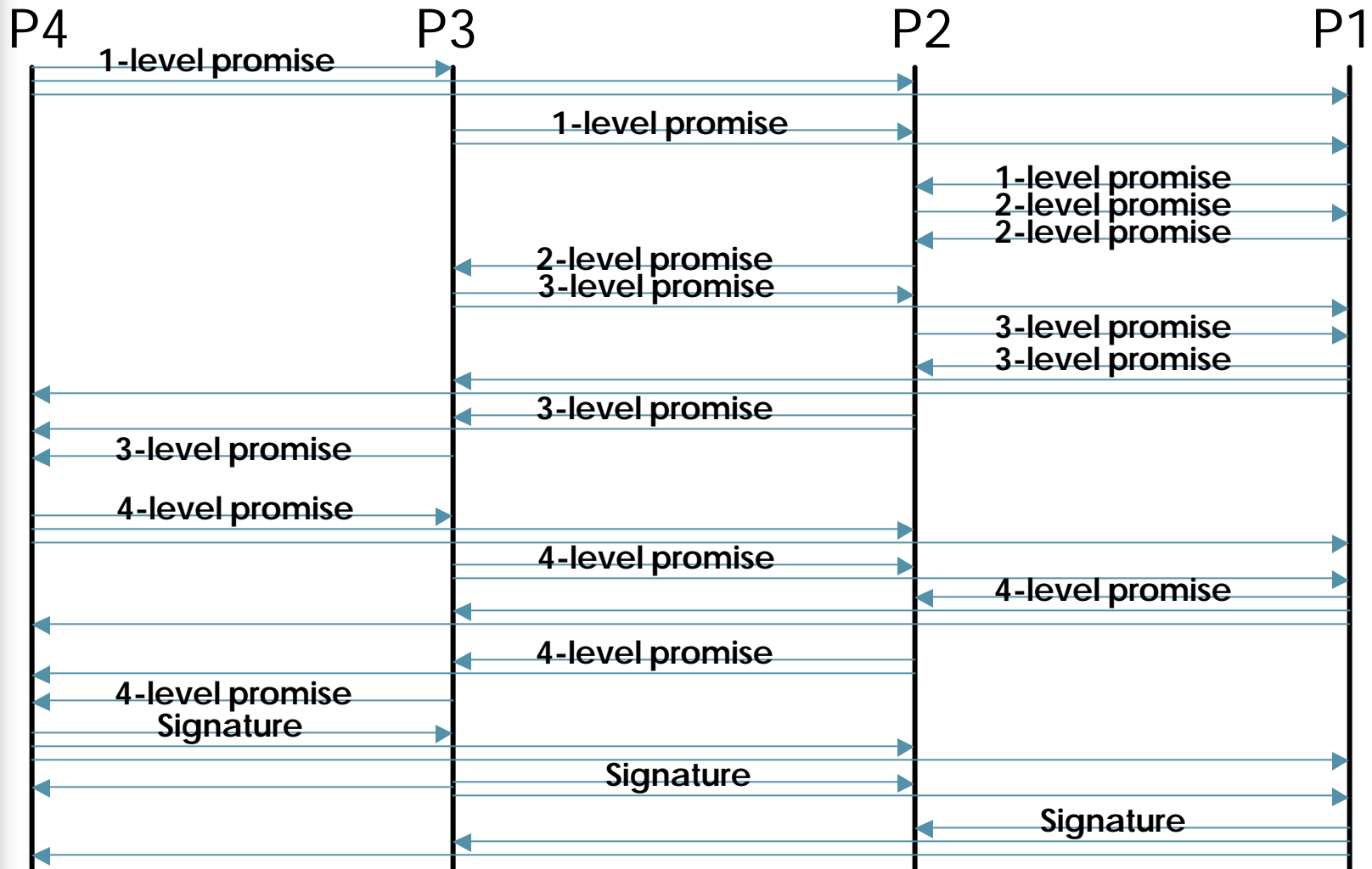
- First formal analysis of multi-party contract signing protocols
- Using the model-checker Mocha and the logic ATL instances of two protocols have been verified
- Two new attacks have been discovered in the GM protocol
  - Abuse-freeness can be broken using side information given by T: easy to fix
  - Fairness can be broken when  $n > 3$ : requires major changes to be fixed



# Future work

- Extend strand space formalism to model fair exchange protocols
  - derive Mocha specifications directly from strands
  - correctness proofs when no attack is found
- Extend the analysis to a more complete model
  - Dolev-Yao-like intruder
  - Parametric verification
- Study different topologies, e.g. ring topologies in fair exchange
- Model optimistic players in multi-party protocols
- Extend general results on advantage, presented in [Chadha, Mitchell, Scedrov, Shmatikov 2003] to multi-party protocols

# GM main prot. (4 participants)



# GM main protocol for $P_i$ (detailed)

