# Round-optimal and Abuse-Free Optimistic Multi-party Contract Signing

**Birgit Baum-Waidner, Entrust Technologies, Zürich**
**Michael Waidner, IBM Zurich Research Laboratory**

**ICALP 2000**
**Geneva, July 12th, 2000**

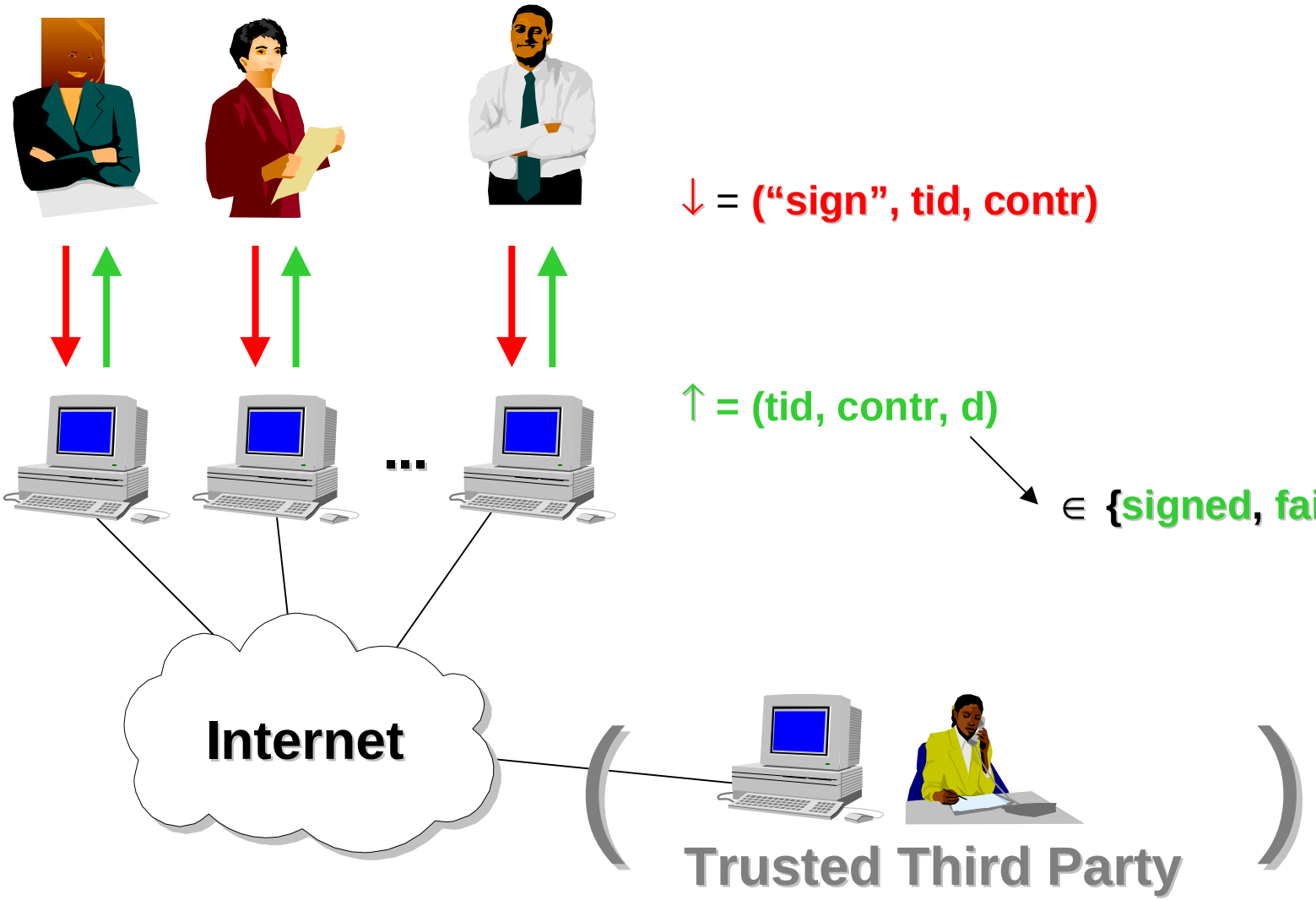
# Outline

1. Introduction
2. Round-optimal optimistic multi-party contract signing
3. Abuse freeness
4. Summary

# Outline

1. Introduction
2. Round-optimal optimistic multi-party contract signing
3. Abuse freeness
4. Summary

# 1.1 Scenario for contract signing



# 1.2 Model

## Participants

- Signatories $P_1, ..., P_n$ $\text{sign}_i(m)$
- Verifiers $V_1, ..., V_{n'}$
- Trusted third party $T$ $\text{sign}_T(m)$
- Can non-deterministically decide to stop waiting for events

## Adversary model

- Up to *t* of the *n* signatories are maliciously dishonest
- Signatures are unforgeable

## Network

- Asynchronous, scheduled by adversary
- Eventual delivery, but in arbitrary order

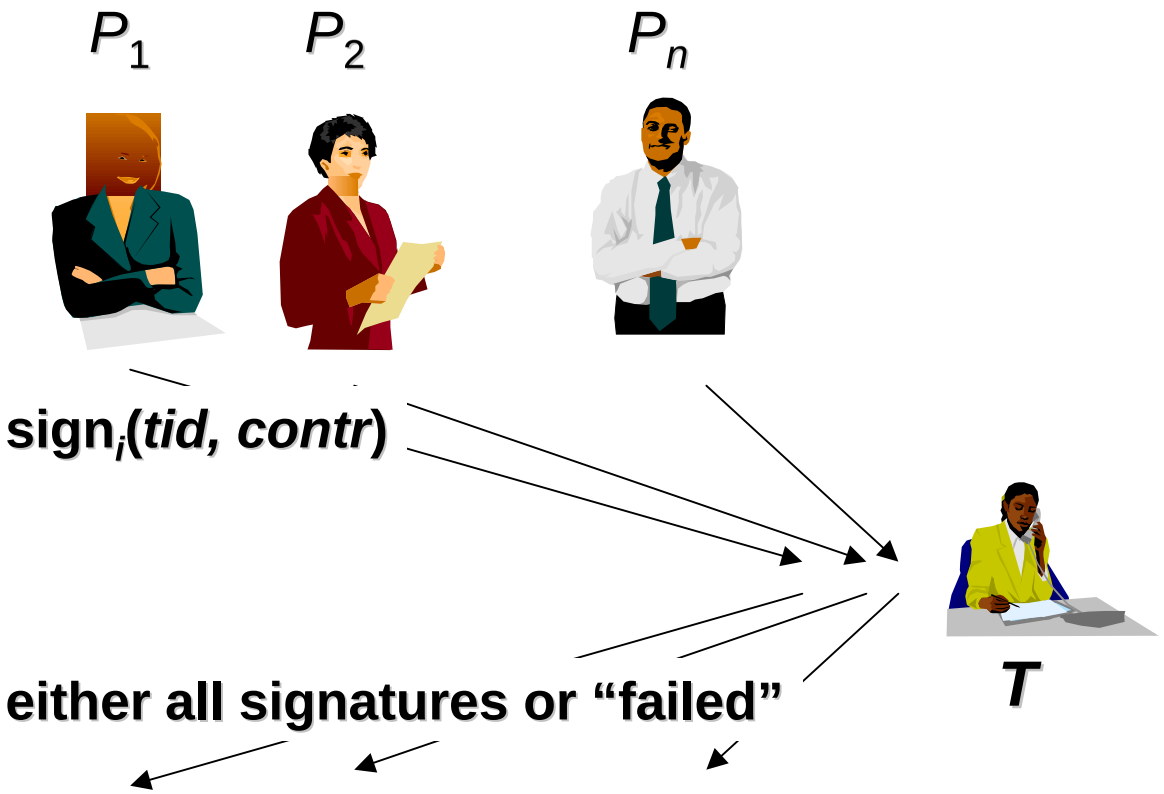
# 1.3 Security requirements

- **Correctness**
  - all $P_i$ are honest, 'patient' and sign $\Rightarrow$ all decide "signed"
- **Unforgeability & no surprises with invalid contracts**
  - $P_i$ never signed, or decided "failed" $\Rightarrow$ $V_j$ never decides "signed"
- **Verifiability**
  - $P_i$ decided "signed," $V_j$ is 'patient' $\Rightarrow$ $V_j$ decides "signed"
- **Termination**

- **Abuse freeness**
  - Adversary cannot prove to an outside party that he has the power to control the contract [Garay, Jakobsson, MacKenzie 1999]

# 1.4 State of the art

## Naive protocol with inline third party:




[Rabin 1981] for 2 party case

# 1.4 State of the art

## Avoid *T* as trust bottleneck

- Gradual exchange of privilege, i.e., exchange fragments of contract, in many rounds
- Limitation: error at least linear in number of rounds

[Ben Or, Goldreich, Micali, Rivest 1990]

## Minimize involvement of *T*

- **Optimistic contract signing,** i.e., involve third party only if something goes wrong

[Micali 1997; Asokan, Schunter, Waidner 1997]

# 1.4 State of the art

| | **Synchronous optimistic** | **Asynchronous optimistic** | **Async. opt. + abuse free** |
|---|---|---|---|
| ***n* = 2** | Micali 1997, Asokan, Schunter, Waidner 1997 | Asokan, Shoup, Waidner 1998 | Garay, Jakobsson, MacKenzie 1999 |
| **any *n*** | Asokan, Schunter, Waidner 1999 | Baum-Waidner, Waidner 1998 | Garay, MacKenzie 1999 |

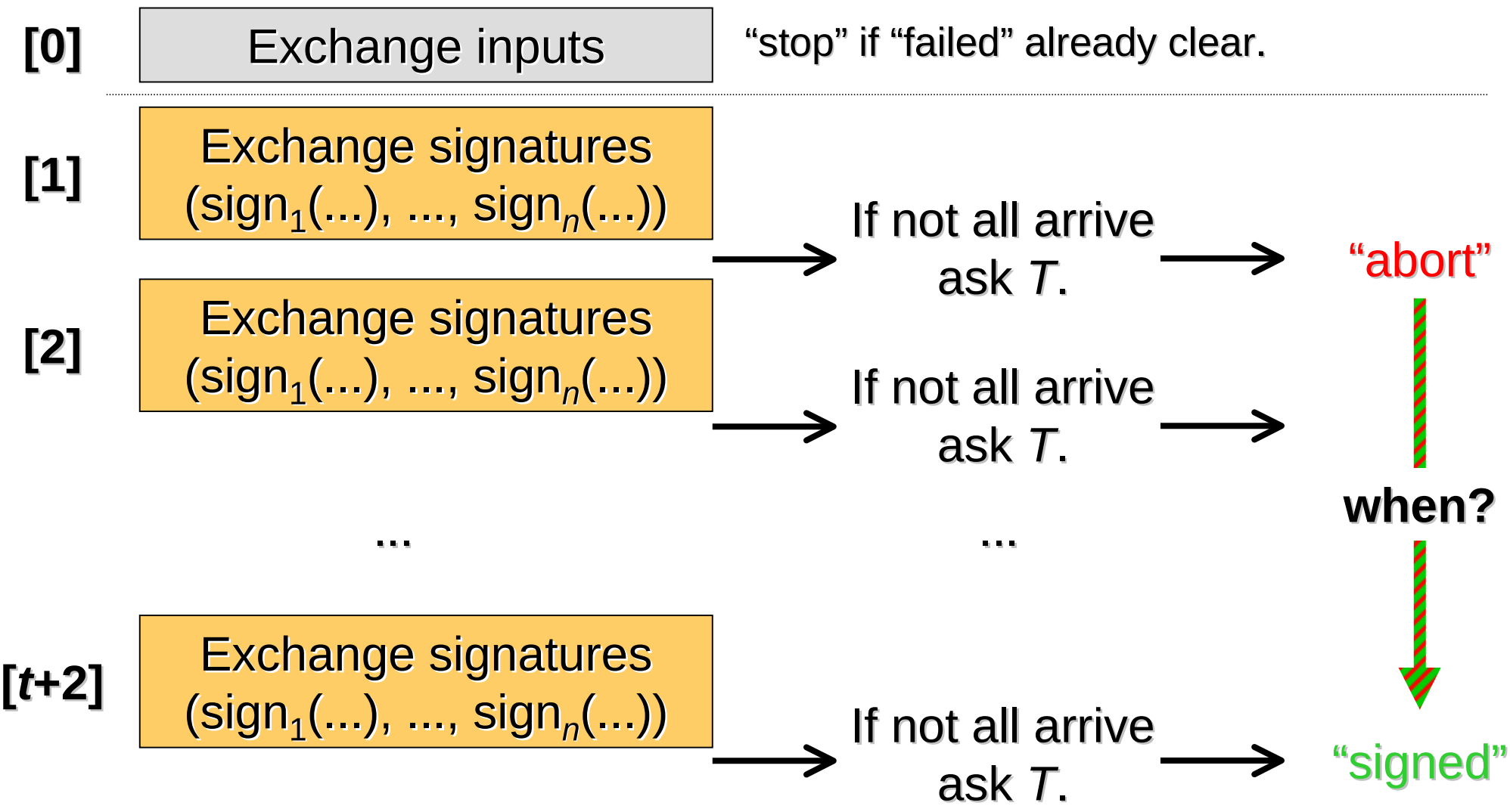
## Our contributions:

- Optimally efficient protocol for asynchronous, abuse-free contract signing: factor *n* over state of the art
- Provably secure construction for abuse freeness, using standard cryptographic operations only

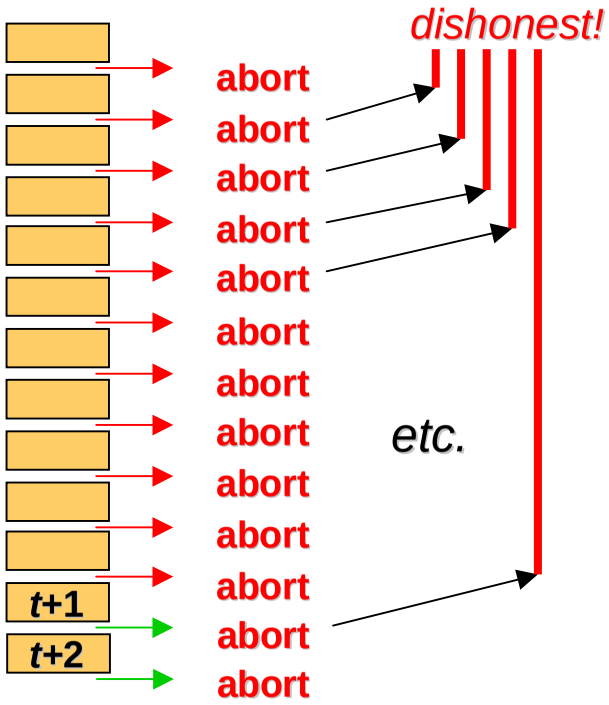
# Outline

1. Introduction
2. Round-optimal optimistic multi-party contract signing
3. Abuse freeness
4. Summary

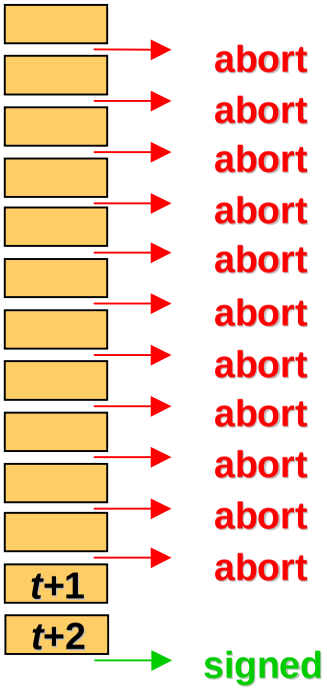
# 2.1 Protocol outline



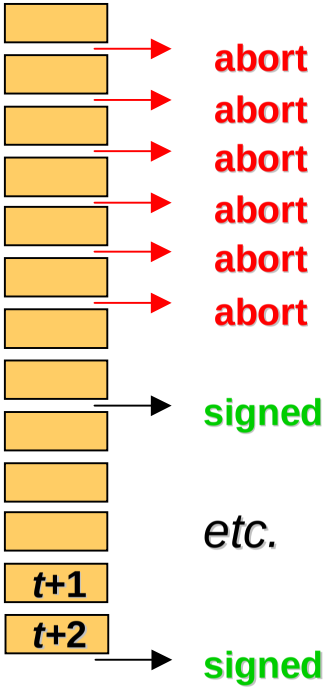
# 2.2 When to switch from “abort” to “signed”?




**Case 1**

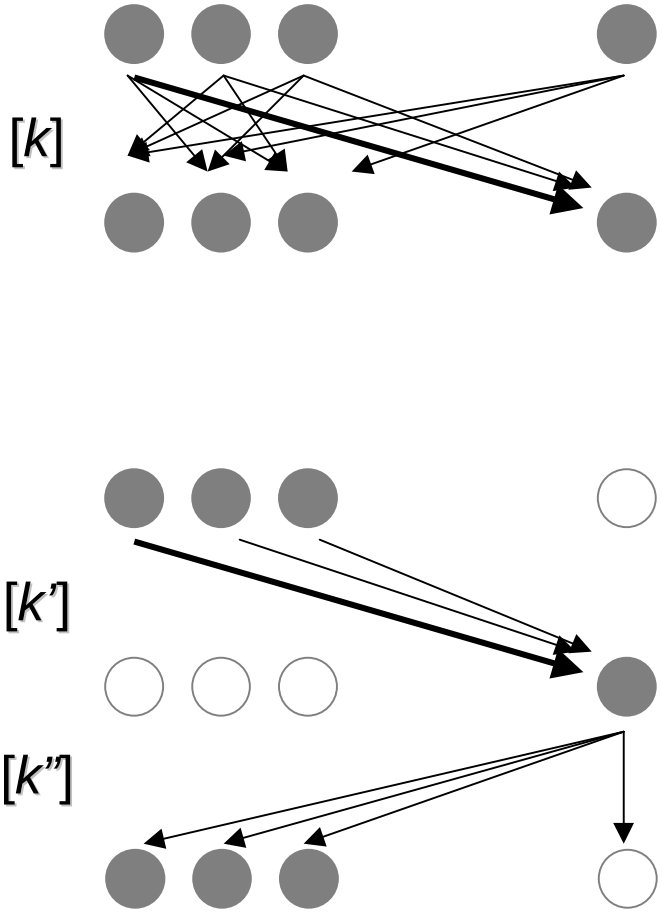



**Case 2**




**General case**

# 2.3 Reduce messages




"Exchange" round with $n(n\text{-}1)$ messages.

2 rounds with $(n\text{-}1)$ messages each.

# 2.4 Complexity

## Worst case

- Round 0: Exchange = 2 rounds with O($n$) messages
- Rounds 1, ..., $t$+2: Exchange = 2 rounds w/ O($n$) msg
- Each party might contact $T$: +2 rounds with O($n$) msg

## Total

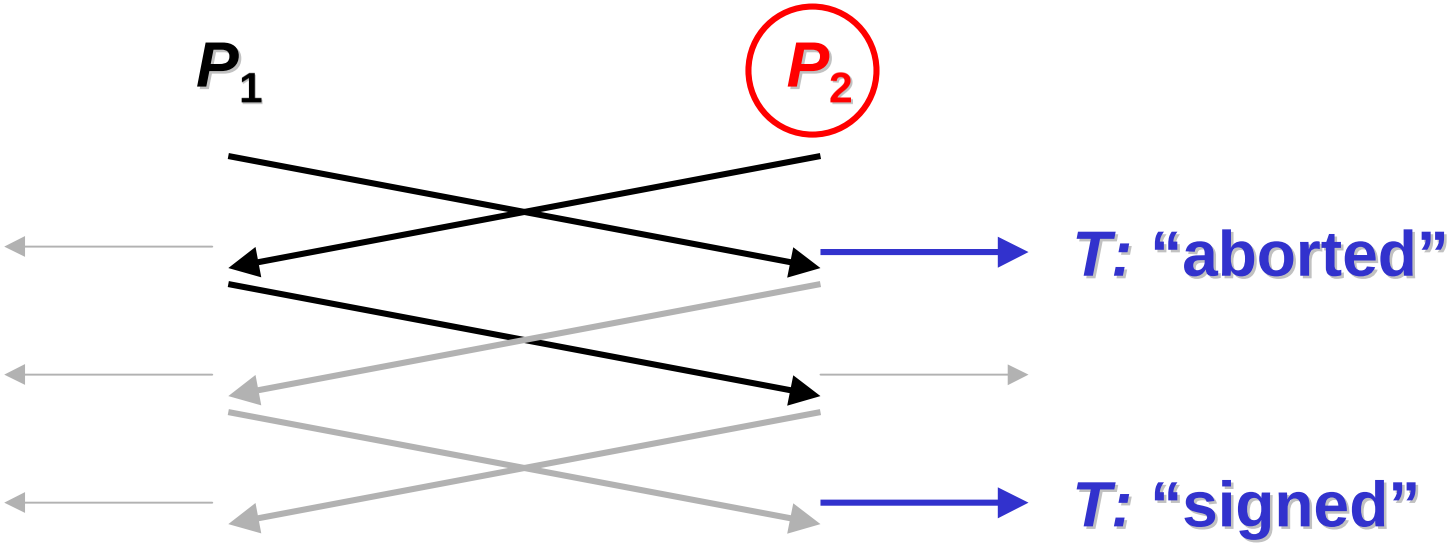
- 2$t$+8 rounds (or $t$+5 rounds)
- O($tn$) messages (or O($tn^2$) messages)
- O($tn^2$ log $n$) bit

# Outline

1. Introduction
2. Round-optimal optimistic multi-party contract signing
3. Abuse freeness
4. Summary

# 3.1 Abuse freeness

## Basic protocol allows “abuse:”



# 3.2 Abuse-free protocol

Distribute fresh keys $pk_i$ + encrypted "certificates" $ecert_i$:

$cert_i := \mathrm{SIGN}_i(i, pk_i, tid, contr)$ (kept secret!)

$ecert_i := E_T(r_i; cert_i)$

Use fresh keys to sign "pre-contract"

$(tid, contr, pk_1, ..., pk_n, ecert_1, ..., ecert_n)$

If pre-contract is signed then open all $ecert_i$

(if necessary with *T*'s help)

signed contract = signed pre-contract +

$(r_1, cert_1, ..., r_n, cert_n)$ or $\mathrm{sign}_T(cert_1, ..., cert_n)$

# 3.2 Abuse-free protocol

Distribute fresh keys $pk_i$ + $ecert_i$:

$cert_i$ := $SIGN_i(i, pk_i, tid, contr)$
$ecert_i$ := $E_T(r_i; cert_i)$

Use fresh keys to sign "pre-contract"
($tid$, $contr$, $pk_1$, ..., $pk_n$, $ecert_1$, ..., $ecert_n$)

**Simulatable until pre-contract is signed.**

---

If pre-contract is signed then open all $ecert_i$
(if necessary with $T$'s help).

signed contract = signed pre-contract +
($r_1$, $cert_1$, ..., $r_n$, $cert_n$) or $sign_T(cert_1, ..., cert_n)$

**If pre-contract is signed then decision on real contract is determined.**

# Outline

1. Introduction
2. Round-optimal optimistic multi-party contract signing
3. Abuse freeness
4. Summary

# 4 Summary

## Result

- Optimistic multi-party contract signing protocol
- Factor of *n* more efficient than previous solution, and provably optimal for $t = n-1$
- Simple solution for abuse-freeness

## Outlook

- Optimally efficient protocols for general *t*
- Reductions of other fair exchange problems to contract signing