Solving Fair Exchange with Mobile Agents

Henning Pagnia Holger Vogt Felix Gärtner
Darmstadt University of Technology, Germany
{pagnia|holgervo|felix}@informatik.tu-darmstadt.de

Uwe G. Wilhelm
Swiss Federal Institute of Technology, Lausanne, Switzerland
Uwe.Wilhelm@epfl.ch

Solving Fair Exchange with Mobile Agents

Henning Pagnia Holger Vogt Felix Gärtner
Darmstadt University of Technology, Germany
{pagnia|holgervo|felix}@informatik.tu-darmstadt.de

Uwe G. Wilhelm
Swiss Federal Institute of Technology, Lausanne, Switzerland
Uwe.Wilhelm@epfl.ch

Mobile agents and fair exchange

- Autonomous agents roam the web and perform electronic business transactions on behalf of the user.
- Items (goods, payment) must be exchanged in a fair manner.
- Fair exchange problem = how to exchange items between two parties without either party suffering a disadvantage.
- Our contribution: three increasingly flexible solutions to the problem using mobile agents.

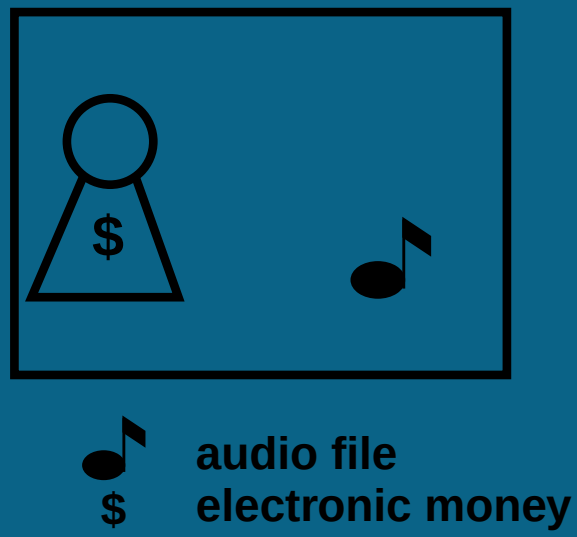
What's the problem with fair exchange?

- An "unfair" exchange protocol:
  1. Agent enters vendor's host.
  2. Agent receives audio file.
  3. Agent pays electronically.
  4. Agent leaves host.

audio file
electronic money

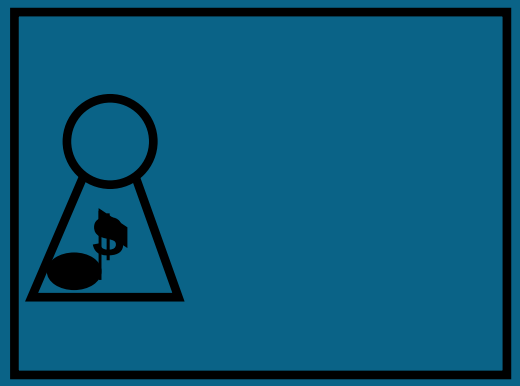
What's the problem with fair exchange?

- An "unfair" exchange protocol:
  1. Agent enters vendor's host.
  2. Agent receives audio file.
  3. Agent pays electronically.
  4. Agent leaves host.



What's the problem with fair exchange?

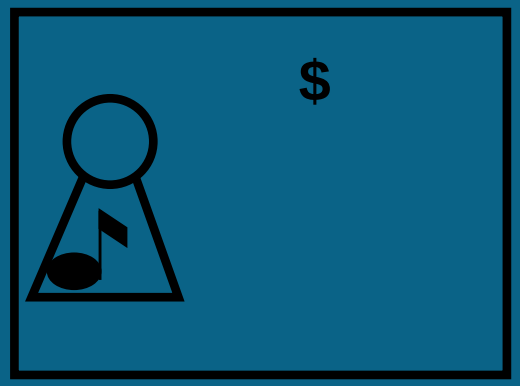
- An "unfair" exchange protocol:
  1. Agent enters vendor's host.
  2. Agent receives audio file.
  3. Agent pays electronically.
  4. Agent leaves host.



audio file
electronic money

What's the problem with fair exchange?

- An "unfair" exchange protocol:
  1. Agent enters vendor's host.
  2. Agent receives audio file.
  3. Agent pays electronically.
  4. Agent leaves host.

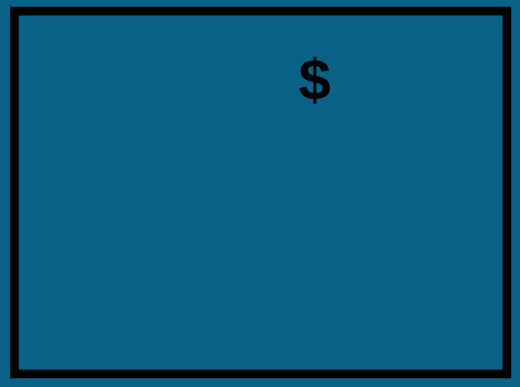


audio file
electronic money

What's the problem with fair exchange?

- An "unfair" exchange protocol:

1. Agent enters vendor's host.
2. Agent receives audio file.
3. Agent pays electronically.
4. Agent leaves host.



audio file
electronic money

- Visiting agent can run without paying (after step 2).
- Vendor can kidnap agent (after step 3).

Solution 1: locked room

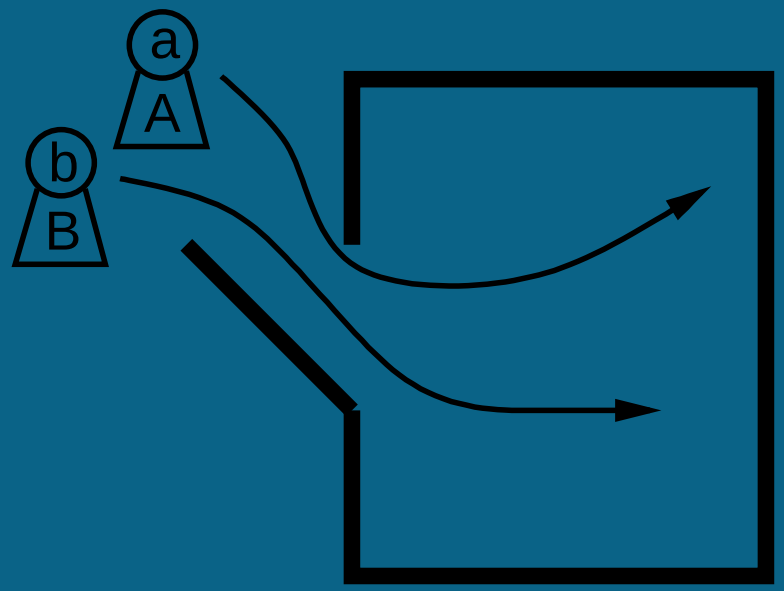
- Locked room protocol:
  1. Agents enter.
  2. Doors close, agents swap.
  3. Agents check and commit.
  4. Doors open, agents leave.

- Ensure that no information leaves the room!

- Ensure that agents are destroyed if one does not commit!

Solution 1: locked room

- Locked room protocol:
  1. Agents enter.
  2. Doors close, agents swap.
  3. Agents check and commit.
  4. Doors open, agents leave.

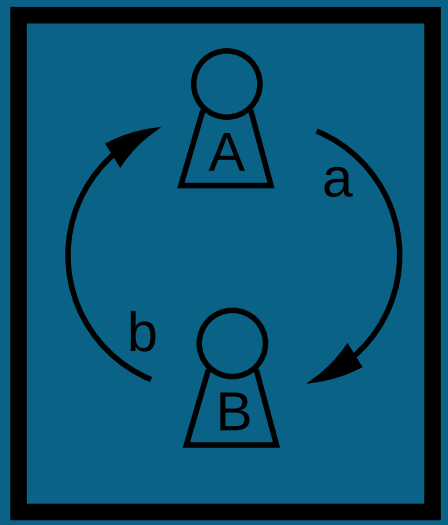


- Ensure that no information leaves the room!

- Ensure that agents are destroyed if one does not commit!

Solution 1: locked room

- Locked room protocol:
  1. Agents enter.
  2. Doors close, agents swap.
  3. Agents check and commit.
  4. Doors open, agents leave.

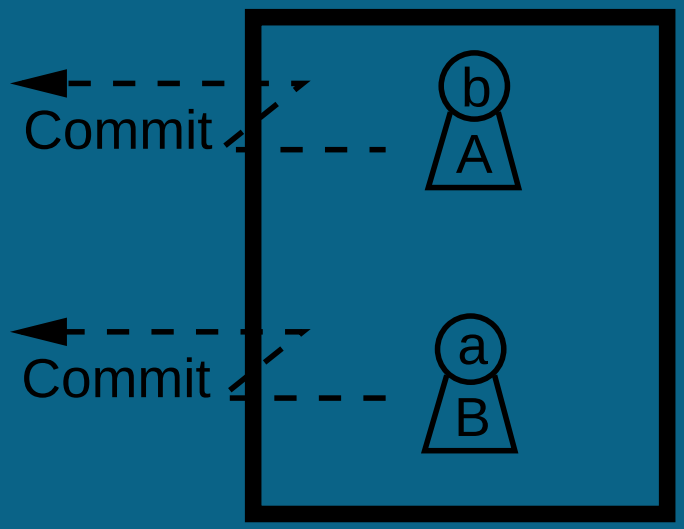


- Ensure that no information leaves the room!

- Ensure that agents are destroyed if one does not commit!

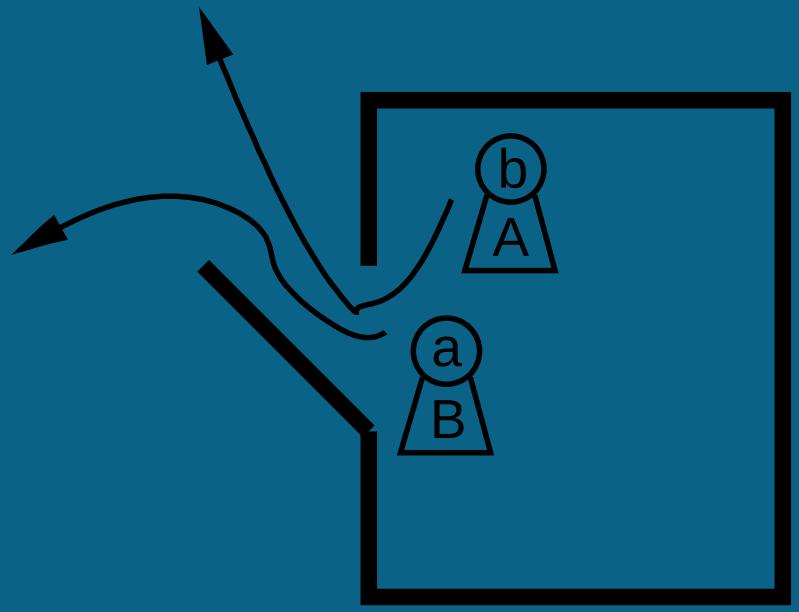
Solution 1: locked room

- Locked room protocol:
  1. Agents enter.
  2. Doors close, agents swap.
  3. Agents check and commit.
  4. Doors open, agents leave.
- Ensure that no information leaves the room!
- Ensure that agents are destroyed if one does not commit!



Solution 1: locked room

- Locked room protocol:
  1. Agents enter.
  2. Doors close, agents swap.
  3. Agents check and commit.
  4. Doors open, agents leave.



- Ensure that no information leaves the room!

- Ensure that agents are destroyed if one does not commit!

Solution 1: locked room

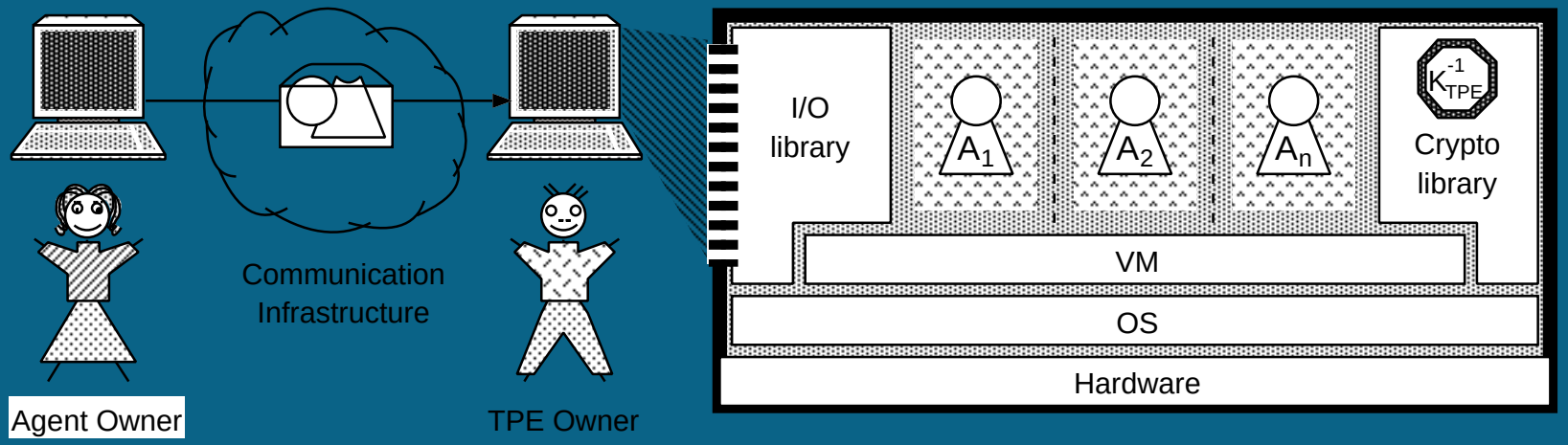
- Locked room protocol:
  1. Agents enter.
  2. Doors close, agents swap.
  3. Agents check and commit.
  4. Doors open, agents leave.

- Ensure that no information leaves the room!

- Ensure that agents are destroyed if one does not commit!

Trusted Processing Environment (TPE)

- Provides secure execution environment on tamper proof hardware device.



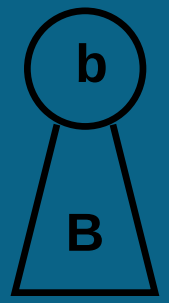
- Protect agents from host and agents from agents.
- Must be fully certified.

Implementation of solution 1

- Protection guarantees formalized as *policies* associated with underlying hardware.
- Implement new *fair exchange policy* based on the following operations:
  - ⋆ `BeginFairExchange(AgentId id)`
  - ⋆ `CommitFairExchange()`
  - ⋆ `AbortFairExchange()`
- TPE restricts communication during exchange and destroys both agents if one doesn't commit.

Solutions 2 & 3: use fair exchange agent

- Use an intermediate *fair exchange agent* (FEA) to validate and swap items.
- FEA performs exchange only if items are as expected.

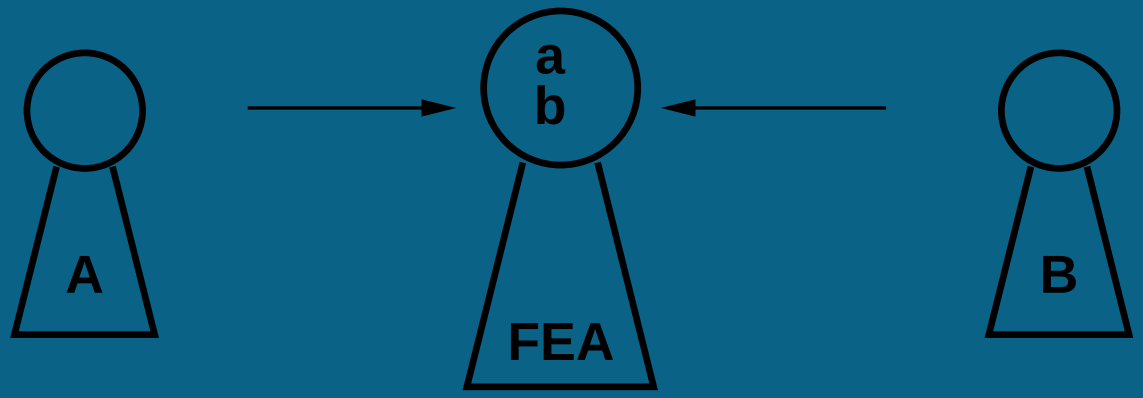


Solutions 2 & 3: use fair exchange agent

- Use an intermediate *fair exchange agent* (FEA) to validate and swap items.
- FEA performs exchange only if items are as expected.

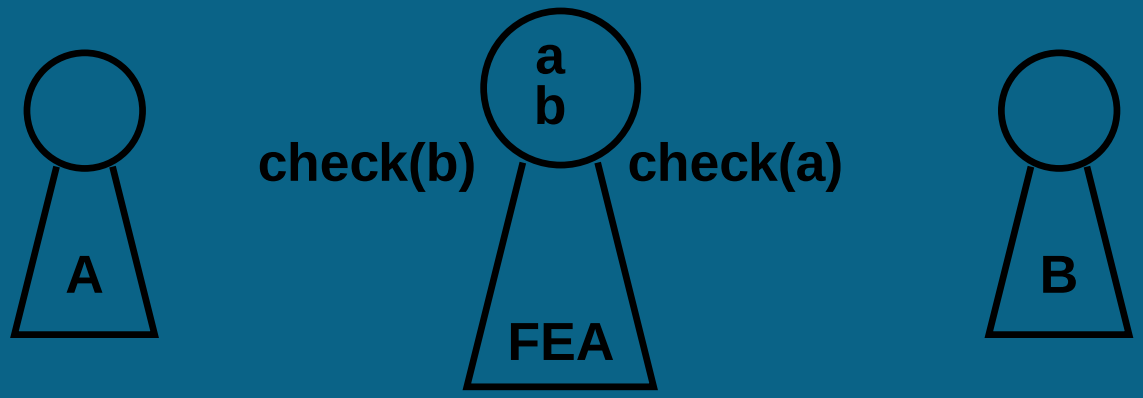
Solutions 2 & 3: use fair exchange agent

- Use an intermediate *fair exchange agent* (FEA) to validate and swap items.
- FEA performs exchange only if items are as expected.



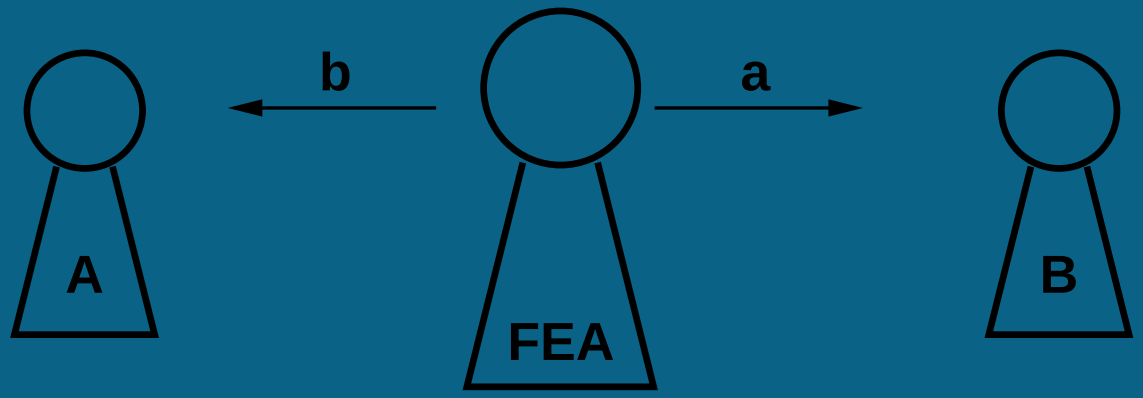
Solutions 2 & 3: use fair exchange agent

- Use an intermediate *fair exchange agent* (FEA) to validate and swap items.
- FEA performs exchange only if items are as expected.



Solutions 2 & 3: use fair exchange agent

- Use an intermediate *fair exchange agent* (FEA) to validate and swap items.
- FEA performs exchange only if items are as expected.



Solutions 2 & 3: use fair exchange agent

- Use an intermediate *fair exchange agent* (FEA) to validate and swap items.
- FEA performs exchange only if items are as expected.

The check routine problem

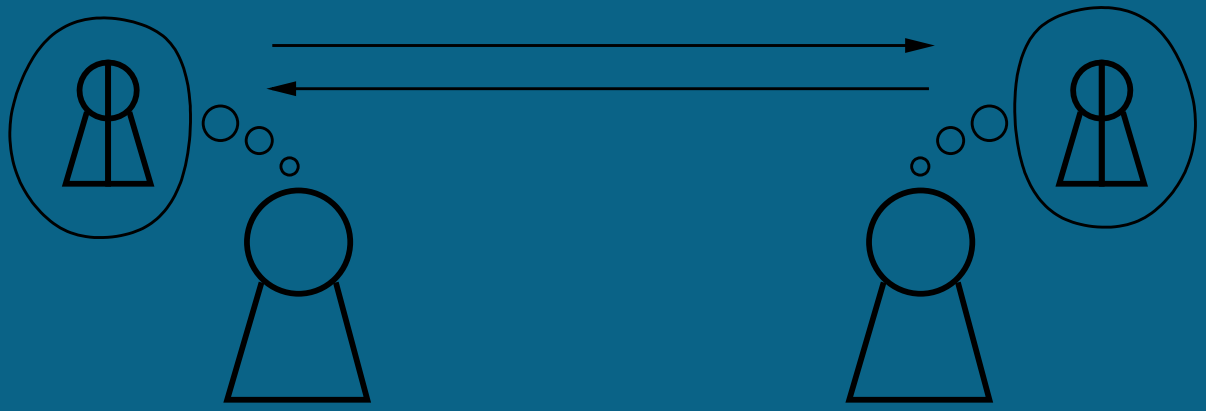
- Validation must be done inside FEA.
- Agents devise specific `check` method.
- Must ensure that no information leaks out of `check` method = check routine problem.
- Possible solutions:
  - ⋆ Parametrized `check` routines.
  - ⋆ Sandboxing.
  - ⋆ . . .

Solution 2

- Let agents check the `check` routines and agree on a mutually checked FEA.
- Agents trust FEA because executed code is ensured to be authentic.
- Only generic TPE-policy of *authentic code* required (no change of TPE necessary).

Solution 2

- Let agents check the `check` routines and agree on a mutually checked FEA.
- Agents trust FEA because executed code is ensured to be authentic.
- Only generic TPE-policy of *authentic code* required (no change of TPE necessary).



Solution 2

- Let agents check the `check` routines and agree on a mutually checked FEA.
- Agents trust FEA because executed code is ensured to be authentic.
- Only generic TPE-policy of *authentic code* required (no change of TPE necessary).

Solution 3

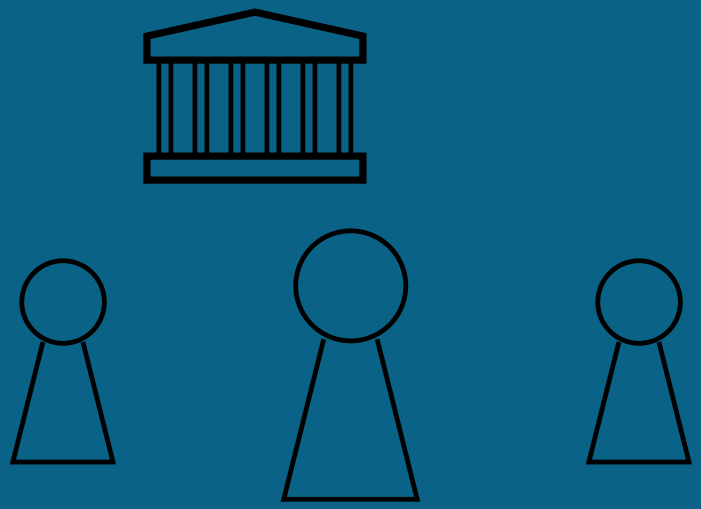
- Use a trusted "free-lance" FEA to perform swap.
- FEA must be certified.
- Only basic TPE functionality required.

Solution 3

- Use a trusted "free-lance" FEA to perform swap.
- FEA must be certified.
- Only basic TPE functionality required.

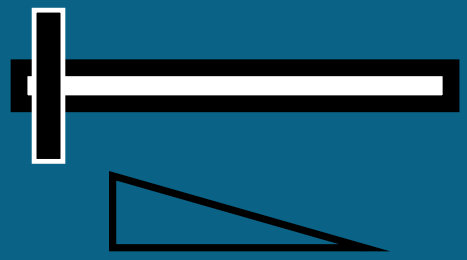
Solution 3

- Use a trusted "free-lance" FEA to perform swap.
- FEA must be certified.
- Only basic TPE functionality required.



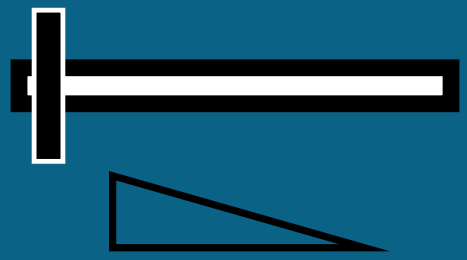
Solution summary and overview

	Solution	Fairness ensured by	Requirements on TPE
1.	Locked room	TPE	Specific fair exchange operations
2.	Authentic code	FEA code checking	generic authentic code
3.	Free-lance FEA	FEA provider	basic protection



Solution summary and overview

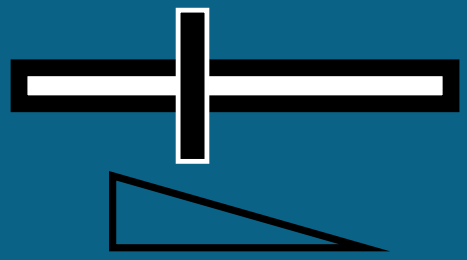
	Solution	Fairness ensured by	Requirements on TPE
1.	Locked room	TPE	Specific fair exchange operations
2.	Authentic code	FEA code checking	generic authentic code
3.	Free-lance FEA	FEA provider	basic protection



Complexity of TPE

Solution summary and overview

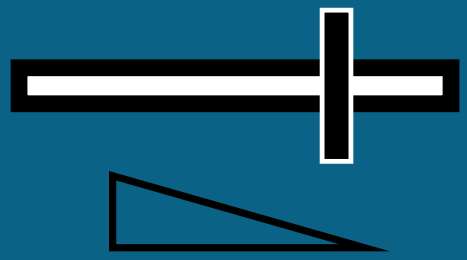
	Solution	Fairness ensured by	Requirements on TPE
1.	Locked room	TPE	Specific fair exchange operations
2.	Authentic code	FEA code checking	generic authentic code
3.	Free-lance FEA	FEA provider	basic protection



Complexity of TPE

Solution summary and overview

	Solution	Fairness ensured by	Requirements on TPE
1.	Locked room	TPE	Specific fair exchange operations
2.	Authentic code	FEA code checking	generic authentic code
3.	Free-lance FEA	FEA provider	basic protection



Advanced questions and future work

- TPE ist still rather “fictional”: IBM 4758 PCI useable?
- Adaption of protocols using other means to ensure security possible, e.g. Smartcards (prior talk by Günter Karjoth)?
- In Solutions 2 & 3 the FEA plays the role of a “trusted third party” (TTP). What constitutes a TTP and where is the TTP in solution 1?

Acknowledgements

- Slides produced using "cutting edge" LaTeX slide processor PPower4 by Klaus Guntermann.