Abstractions for the Formal Analysis of Optimistic Exchange Protocols

[Work in Progress]

A. Boisseau[1], S. Kremer[2] and J.-F. Raskin[2]

[1]Laboratoire Spécification et Vérification (LSV),
CNRS & ENS de Cachan, France
[2]Université Libre de Bruxelles, Department of Computer Science,
Belgium

What are fair exchange protocols ?

- electronic purchase of goods: exchange of an electronic item against an electronic payment
- digital contract signing: exchange of digital signatures on a given electronic document
- non-repudiation protocols: exchange of an electronic item and a nro evidence against the corresponding nrr evidence
- certified e-mail: exchange of an electronic message against a proof of receipt
- ...

Contract Signing Protocols

$$
\begin{array}{lllll}
(1) & A & \rightarrow & B & : \quad SIG_A(C) \\
(2) & B & \rightarrow & A & : \quad SIG_B(C)
\end{array}
$$

Contract Signing Protocols

$$\begin{array}{lcccccc} (1) & A & \rightarrow & B & : & SIG_A(C) \\ (2) & B & \rightarrow & A & : & SIG_B(C) \end{array}$$

Asymmetry implies advantages of B over A:

- B can stop the protocol
- B can influence an external observer by showing A's signature

Contract Signing Protocols

$$
\begin{array}{lllll}
(1) & A & \rightarrow & B & : SIG_A(C) \\
(2) & B & \rightarrow & A & : SIG_B(C)
\end{array}
$$

Asymmetry implies advantages of B over A:

- B can stop the protocol
- B can influence an external observer by showing A's signature

Solution [S. Even, Y. Yacobi – Relations among Public Key Signature Systems]:

- $\left.\begin{array}{l}\text{Probabilistic}\\ \text{Gradual exchange}\end{array}\right\}$ unrealistic assumptions and/or inefficient
- Trusted Third Party (in particular: optimistic protocols)

GJM Protocol

[J. A. Garay, M. Jakobsson, P. Mac Kenzie – Abuse-Free Optimistic Contract Signing]

Exchange :

1) $A \to B : PCS_{A,B,T}(C)$
2) $B \to A : PCS_{B,A,T}(C)$
3) $A \to B : SIG_A(C)$
4) $B \to A : SIG_B(C)$

Resolve(A) :

1) $A \to T : \langle PCS_{B,A,T}(C), SIG_A(C)\rangle$
2) $T \to A : \begin{cases} SIG_T(abort) \text{ if aborted} \\ SIG_B(C) \text{ otherwise} \end{cases}$

Resolve(B) :

1) $A \to T : \langle PCS_{B,A,T}(C), SIG_A(C)\rangle$
2) $T \to A : \begin{cases} SIG_T(abort) \text{ if aborted} \\ SIG_B(C) \text{ otherwise} \end{cases}$

Abort :

1) $A \to T : SIG_A(abort)$
2) $T \to A : \begin{cases} SIG_B(C) \text{ if resolved} \\ SIG_T(abort) \text{ otherwise} \end{cases}$

Specific to fair exchange protocols

- Branching protocols *vs Ping-Pong protocols*
- Competition between participants
  vs Competition between participants and intruder
- Fairness, Timeliness and Abuse-freeness *vs Secret and Authentication*

Expected Properties

- Fairness: *"it is impossible for a participant to obtain a valid contract without allowing the remaining participant to do the same"*
- Timeliness: *"at any moment in the protocol, each participant can reach a point where it can stop the protocol, achieving fairness"*
- Abuse-Freeness: *"it is impossible for a participant, to be able to prove to an external observer that he has the power to determine the outcome of the protocol"*

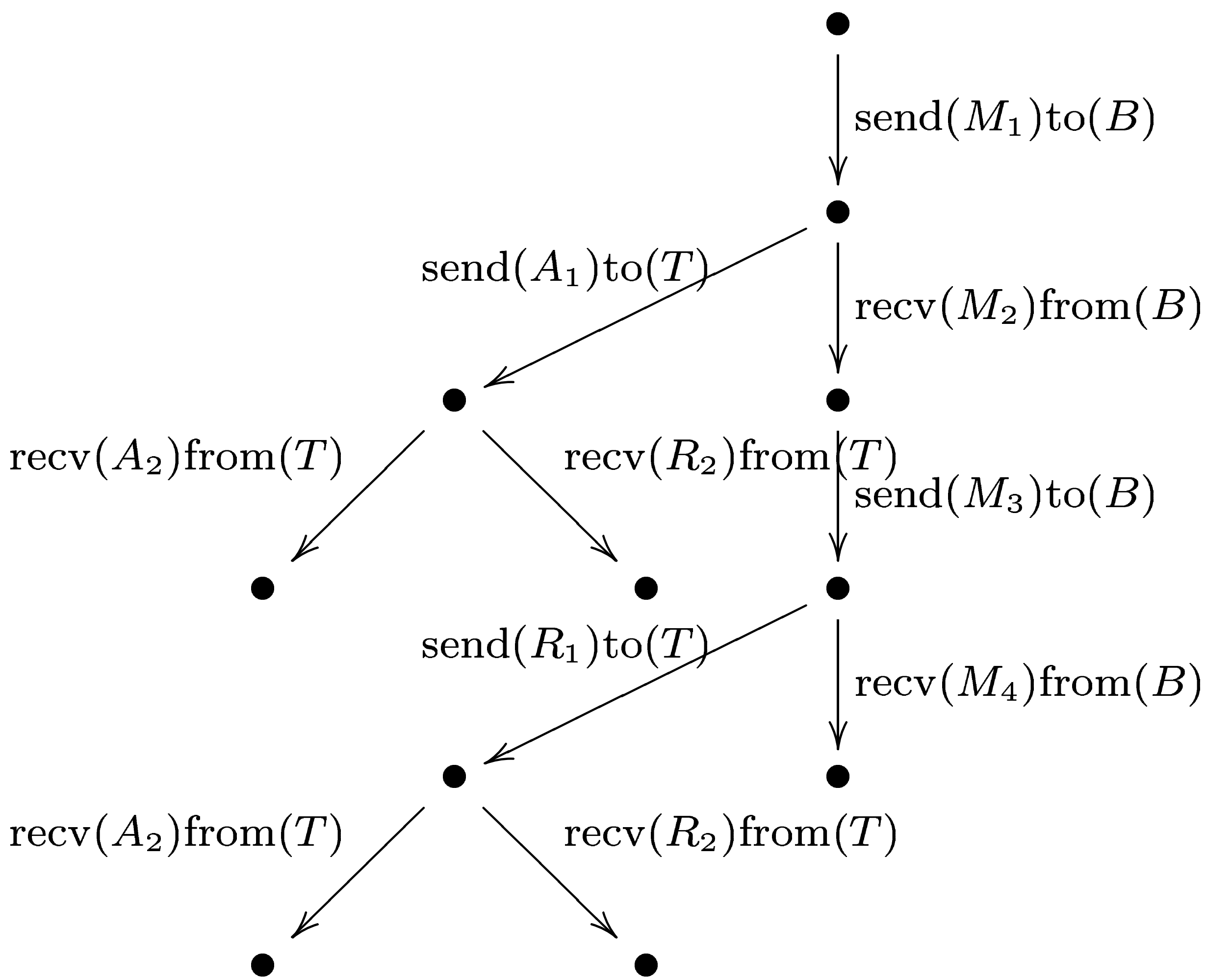
Related Work

- [V. Shmatikov, J. C. Mitchell – Finite State Analysis of Two Contract Signing Protocols]
  Modeling with transition systems
  Verification with Murφ
- [R. Chada, M.Kanovich, A. Scedrov – Inductive Methods and Contract-Signing Protocols]
  Modeling with MSR
  Inductive proofs
- [S. Kremer, J.-F. Raskin – Game Analysis of Abuse-Free Contract Signing]
  Game modeling with ATS and ATL
  Verification with MOCHA

Our Approach

- Game based modeling
  (based on [S. Kremer, J.-F. Raskin – Game Analysis of Abuse-Free Contract Signing])
- one protocol session, but TTP responding to any (valid) request
- Replace simplifications by abstractions
- Automation
  - “high-level” specification
  - abstract $\rightarrow$ finite reactive modules
  - model-check using MOCHA

Protocol Syntax: Roles (common participants)



Protocol Syntax: Roles (server participants)

$$\text{recv}(a_1^t)$$

$$[\neg\text{send}(r_2^t)\text{to}(A) \wedge \neg\text{send}(r_2^t)\text{to}(B)]?\{\text{send}(a_2^t)\text{to}(A) : \text{send}(r_2^t)\text{to}(A)\}$$

$$\text{recv}(r_1^t)$$

$$[\neg\text{send}(a_2^t)\text{to}(A)]?\{\text{send}(r_2^t)\text{to}(A) : \text{send}(a_2^t)\text{to}(A)\}$$

$$\text{recv}(ra_1^t)$$

$$[\neg\text{send}(a_2^t)\text{to}(A)]?\{\text{send}(r_2^t)\text{to}(A) : \text{send}(a_2^t)\text{to}(A)\}$$

$$\text{recv}(rb_1^t)$$

$$[\neg\text{send}(a_2^t)\text{to}(A)]?\{\text{send}(r_2^t)\text{to}(B) : \text{send}(a_2^t)\text{to}(B)\}$$

Protocol Syntax: Participants and Channels

Participants $P = (a, R, IK, level) \in \mathcal{P}$ where:

- a: identity
- R: role
- IK: initial knowledge
- $level \in \{honest, weakly, strongly\}$: honesty level

Channels $C = (s, r, level) \in \mathsf{Com}$ where:

- s and r: identities of sender and receiver
- $level \in \{operational, resilient, unreliable\}$: reliability level

ATS

[R. Alur, A. Henzinger, O. Kupferman – Alternating-Time Temporal Logic]

$$S = (\Sigma, Q, q_0, \delta, \pi, \Pi)$$

- Σ: finite set of agents
- For each $a \in \Sigma$:
  - Q_a: set of a's local states

    $$Q = \prod Q_a \text{ and } q_0 \in Q$$

  - $\delta_a : Q \to 2^{Q_a}$: a's local transition function

    $$q \xrightarrow{\delta} q' \;\Leftrightarrow\; \forall a \in \Sigma.\ q \xrightarrow{\delta_a} q'_a$$

- Π: set of atomic propositions
- $\pi : Q \to 2^{\Pi}$: valuation

ATL

[R. Alur, A. Henzinger, O. Kupferman – Alternating-Time Temporal Logic]

An ATL-formula φ is one of the following :

- $p \in \Pi$
- $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$
- $\langle\langle A \rangle\rangle \bigcirc \varphi$, $\langle\langle A \rangle\rangle \varphi_1 \,\mathcal{U}\, \varphi_2$, $\langle\langle A \rangle\rangle \Box \varphi$, $\langle\langle A \rangle\rangle \Diamond \varphi$
- $[\![A]\!] \bigcirc \varphi$, $[\![A]\!] \varphi_1 \,\mathcal{U}\, \varphi_2$, $[\![A]\!] \Box \varphi$, $[\![A]\!] \Diamond \varphi$

Protocol Semantics

Define local states of the ATS:

- each $P \in \mathcal{P}$ is an agent with:
  - Q_P records:
    - Sent and received messages
    - [Stopped or not]
- each $C \in \mathrm{Com}$ is an agent with:
  - Q_C records
    - transmitted messages

Protocol Semantics: Honesty and Reliability

Define local transition function depending on:

- Honesty levels:
  - Honest: Respects the protocol
  - Weakly dishonest: Accepts and forges new messages
  - Strongly dishonest: Spy communications
- Reliability levels:
  - Operational: Messages sent are delivered immediately
  - Resilient: Messages sent are delivered after a finite unknown amount of time
  - Unreliable: Messages can be lost

Formal Properties

- Fairness (for B):

$$\neg \langle\langle A, T, \mathsf{Com} \rangle\rangle \Diamond (contract_A \wedge \neg \langle\langle B \rangle\rangle \Diamond contract_B)$$

- Timeliness (for B):

$$\langle\langle B \rangle\rangle \Diamond ((B\_stop \wedge (contract_B \vee \neg \langle\langle A, T, \mathsf{Com} \rangle\rangle \Diamond contract_A))$$

- Abuse-Freeness (for A):

$$\begin{aligned} \neg \langle\langle A, T, \mathsf{Com} \rangle\rangle \Diamond \quad & (involved_A \wedge \\ & ((\langle\langle A, T, \mathsf{Com} \rangle\rangle \Diamond \quad (aborted_A \wedge \\ & \qquad\qquad\qquad\qquad (\neg \langle\langle B \rangle\rangle \Diamond contract_B))) \wedge \\ & ((\langle\langle A, T, \mathsf{Com} \rangle\rangle \Diamond (contract_A))) \end{aligned}$$

Problem

- Infinite ATS
  - infinite set of messages that can be constructed by a malicious participant:
    infinite set of nonces, ciphers or hashes that cannot be checked, requests to TTP, . . .

- Model-checking not applicable

Solution: use of abstractions

Abstraction for ATS

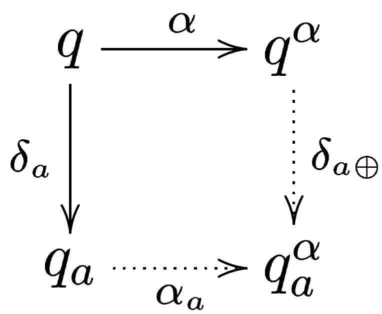
[T. A. Henzinger, R. Majumdar, F. Mang, J.-F. Raskin Abstract Interpretation of Game Properties]

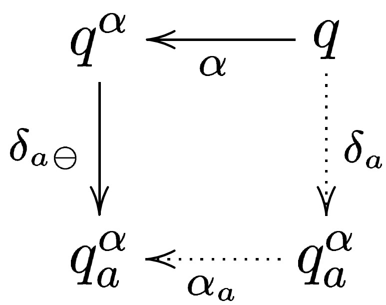
- Principle: abstract δ_a by $(\delta_{a\oplus}, \delta_{a\ominus})$ where:
  $\delta_{a\oplus}$ (resp. $\delta_{a\ominus}$) gives more (resp. less) power to a

Abstraction for ATS

[T. A. Henzinger, R. Majumdar, F. Mang, J.-F. Raskin Abstract Interpretation of Game Properties]

- Principle: abstract δ_a by $(\delta_{a\oplus}, \delta_{a\ominus})$ where:
  $\delta_{a\oplus}$ (resp. $\delta_{a\ominus}$) gives more (resp. less) power to a
- Precisely:
  - surjections $\alpha_a : Q_a \to Q_a^\alpha$
  - $Q^\alpha = \prod Q_a^\alpha$ and $\alpha = \prod \alpha_a$
  - $(\delta_{a\oplus}, \delta_{a\ominus})$ such that:

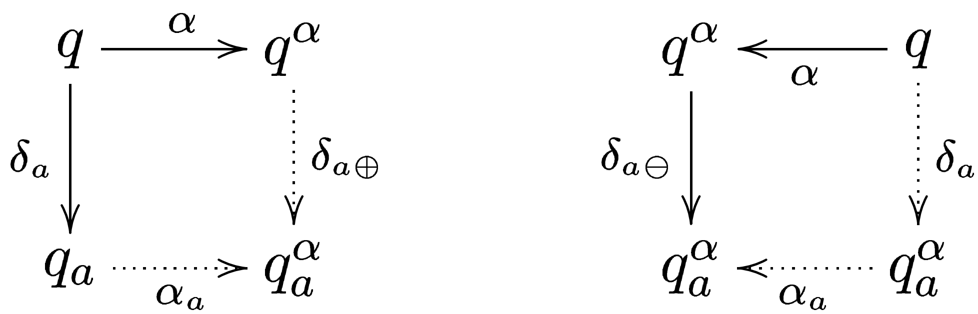




Abstraction for ATS

[T. A. Henzinger, R. Majumdar, F. Mang, J.-F. Raskin Abstract Interpretation of Game Properties]

- Principle: abstract δ_a by $(\delta_{a\oplus}, \delta_{a\ominus})$ where: $\delta_{a\oplus}$ (resp. $\delta_{a\ominus}$) gives more (resp. less) power to a
- Precisely:
  - surjections $\alpha_a : Q_a \rightarrow Q_a^\alpha$
  - $Q^\alpha = \prod Q_a^\alpha$ and $\alpha = \prod \alpha_a$
  - $(\delta_{a\oplus}, \delta_{a\ominus})$ such that:

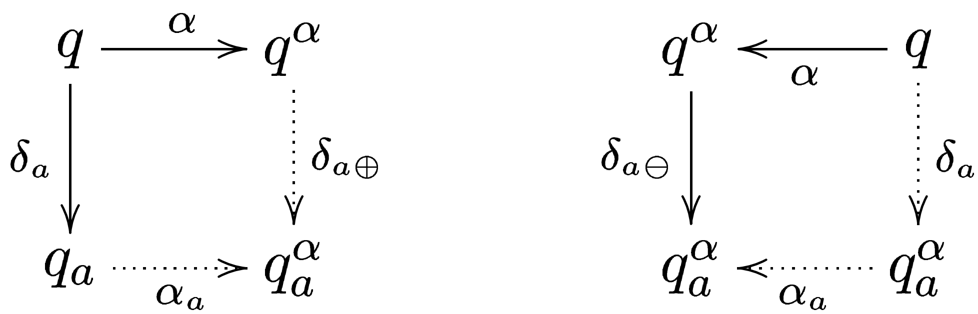


- Other conditions about initial states and valuation

Abstraction for ATS

[T. A. Henzinger, R. Majumdar, F. Mang, J.-F. Raskin Abstract Interpretation of Game Properties]

- Principle: abstract δ_a by $(\delta_{a\oplus}, \delta_{a\ominus})$ where:
  $\delta_{a\oplus}$ (resp. $\delta_{a\ominus}$) gives more (resp. less) power to a
- Precisely:
  - surjections $\alpha_a : Q_a \to Q_a^\alpha$
  - $Q^\alpha = \prod Q_a^\alpha$ and $\alpha = \prod \alpha_a$
  - $(\delta_{a\oplus}, \delta_{a\ominus})$ such that:



- Other conditions about initial states and valuation
- Notation $S^\alpha[A, \Sigma \setminus A]$ for $A \subseteq \Sigma$

Abstractions for ATL

- Allow negation only at propositional level
- To verify:
  $S \models \langle\langle A \rangle\rangle\, \phi$ use $S^\alpha[A, \Sigma \setminus A]$
  $S \models [\![A]\!]\, \phi$ use $S^\alpha[\Sigma \setminus A, A]$

Correctness result:

if $S^\alpha \models \varphi$ then $S \models \varphi$

Abstractions for Protocols

- Concrete semantics contains an infinite set of ground messages
- Avoid ground messages: message patterns + “symbolic substitution” on variables
- Define abstraction function $\alpha : Q \rightarrow Q^{\alpha}$
- Given a protocol description, an abstraction
  Compute the abstract semantics of the protocol

Implementation

[http://www-cad.eecs.berkeley.edu/~tah/mocha/]

- Given a protocol specification, an abstraction
  Compute a set of MOCHA modules
  Implementing the abstract semantics of the protocol
- Given an ATL-formula
  Compute a MOCHA script to check it

Conclusion and Future Work

- Precise semantics for optimistic fair exchange protocol
- Rigorous reduction to a finite, sufficiently small, model
- Implementation

Future work:

- Extension to multi-session
- Extension to multi-party protocols