CS 259

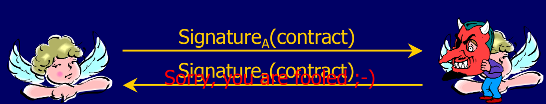
Game-Based Verification of Fair Exchange Protocols

Vitaly Shmatikov

Overview

- Fair exchange protocols
  - Protocols as games
  - Security as presence or absence of certain strategies
- Alternating transition systems
  - Formal model for adversarial protocols
- Alternating-time temporal logic
  - Logic for reasoning about alternating transition systems
- Game-based verification of fair exchange
  - Example: Garay-Jakobsson-MacKenzie protocol

The Problem of Fair Exchange

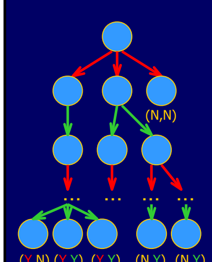


- Malicious participant vs. external intruder
  - Fair exchange protocols are designed to provide protection against misbehavior by protocol participants
- A protocol can be viewed as a game
  - Adversarial behavior (e.g., Alice vs. Bob)
  - Cooperative behavior (e.g., Bob controls communication channel)

Game-Theoretic Model

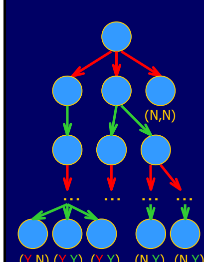
- Each protocol message is a game move
  - Different sets of moves for different participants
- Four possible outcomes (for signature exchange)
  - A has B's signature, B has A's signature
  - A has B's signature, B doesn't have A's signature, etc.
- Honest players follow the protocol
- Dishonest players can make any Dolev-Yao move
  - Send any message they can compute
  - Wait instead of responding
- Reason about players' game strategies

Protocol as a Game Tree



- Every possible execution of the protocol is a path in the tree
- Players alternate their moves
  - First A sends a message, then B, then A …
  - Adversary "folded" into dishonest player
- Every leaf labeled by an outcome
  - (Y,Y) if A has B's signature and B has A's
  - (Y,N) if only A has B's signature, etc.
- Natural concept of strategy
  - A has a strategy for getting B's signature if, for any move B can make, A has a response move such that the game always terminates in some leaf state labeled (Y,…)

Define Properties on Game Trees



Fairness

No leaf node is labeled (Y,N) or (N,Y)

Balance (for A)

B never has a strategy to reach (Y,Y) AND a strategy to reach (N,N)

Abuse-freeness (for A)

B cannot prove that it has advantage

- Not trace-based properties (unlike secrecy and authentication)
- Very difficult to verify with symbolic analysis or process algebras

Alternating Transition Systems

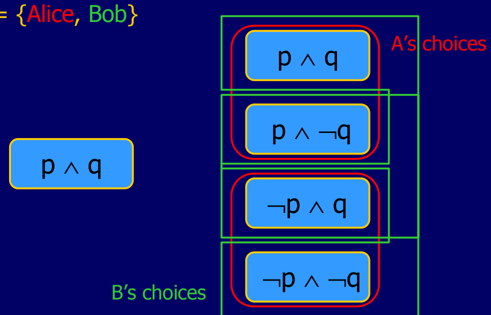
- Game variant of Kripke structures
  - *R. Alur, T. Henzinger, O. Kupferman. "Alternating-time temporal logic". FOCS 1997.*
- Start by defining state space of the protocol
  - Π is a set of propositions
  - Σ is a set of players
  - Q is a set of states
  - $Q_0 \subseteq Q$ is a set of initial states
  - $\pi: Q \to 2^\Pi$ maps each state to the set of propositions that are true in the state
- So far, this is very similar to Murφ

Transition Function

- $\delta: Q \times \Sigma \to 2^{2^Q}$ maps a state and a player to a nonempty set of choices, where each choice is a set of possible next states
  - When the system is in state q, each player chooses a set $Q_a \in \delta(q,a)$
  - The next state is the intersection of choices made by all players $\cap_{a \in \Sigma} \delta(q,a)$
  - The transition function must be defined in such a way that the intersection contains a unique state
- Informally, a player chooses a set of possible next states, then his opponents choose one of them

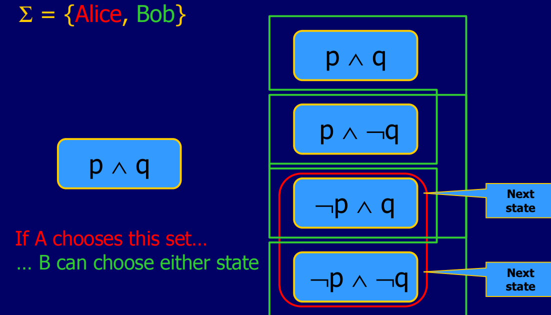
Example: Two-Player ATS

Σ = {Alice, Bob}



Example: Computing Next State

Σ = {Alice, Bob}



Alternating-Time Temporal Logic

- Propositions $p \in \Pi$
- $\neg\varphi$ or $\varphi_1 \vee \varphi_2$ where $\varphi, \varphi_1, \varphi_2$ are ATL formulas
- $\langle\langle A \rangle\rangle \bigcirc \varphi$, $\langle\langle A \rangle\rangle \Box \varphi$, $\langle\langle A \rangle\rangle \varphi_1 U \varphi_2$ where $A \subseteq \Sigma$ is a set of players, $\varphi, \varphi_1, \varphi_2$ are ATL formulas
  - These formulas express the ability of coalition A to achieve a certain outcome
  - $\bigcirc$, $\Box$, U are standard temporal operators (similar to what we saw in PCTL)
- Define $\langle\langle A \rangle\rangle \Diamond \varphi$ as $\langle\langle A \rangle\rangle$ true U φ

Strategies in ATL

- A strategy for a player $a \in \Sigma$ is a mapping $f_a: Q^+ \to 2^Q$ such that for all prefixes $\lambda \in Q^*$ and all states $q \in Q$, $f_a(\lambda \cdot q) \in \delta(q,a)$
  - For each player, strategy maps any sequence of states to a set of possible next states
- Informally, the strategy tells the player in each state what to do next
  - Note that the player cannot choose the next state. He can only choose a set of possible next states, and opponents will choose one of them as the next state.

Temporal ATL Formulas (I)

- ⟨⟨A⟩⟩○φ iff there exists a set F_a of strategies, one for each player in A, such that for all future executions λ∈out(q,F_a) φ holds in first state λ[1]
  - Here out(q,F_a) is the set of all future executions assuming the players follow the strategies prescribed by F_a, i.e., $\lambda = q_0 q_1 q_2 \ldots \in out(q,F_a)$ if $q_0=q$ and $\forall i\ q_{i+1} \in \cap_{a\in A} f_a(\lambda[0,i])$
- Informally, ⟨⟨A⟩⟩○φ holds if coalition A has a strategy such that φ always holds in the next state

Temporal ATL Formulas (II)

- ⟨⟨A⟩⟩□φ iff there exists a set F_a of strategies, one for each player in A, such that for all future executions λ∈out(q,F_a) φ holds in all states
  - Informally, ⟨⟨A⟩⟩□φ holds if coalition A has a strategy such that φ holds in every execution state
- ⟨⟨A⟩⟩◇φ iff there exists a set F_a of strategies, one for each player in A, such that for all future executions λ∈out(q,F_a) φ eventually holds in some state
  - Informally, ⟨⟨A⟩⟩◇φ holds if coalition A has a strategy such that φ is true at some point in every execution

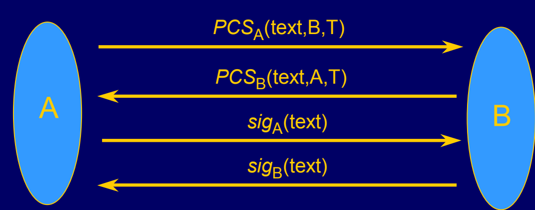
Protocol Description Language

- Guarded command language
  - Very similar to PRISM input language (proposed by the same people)
- Each action described as [] guard → command
  - guard is a boolean predicate over state variables
  - command is an update predicate, same as in PRISM
  - Simple example:

```
[]SigM1B ∧ ¬SendM2 ∧ ¬StopB -> SendMrB1':=true;
```

Abuse-Free Contract Signing

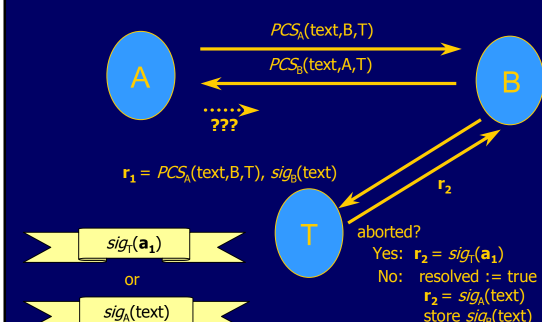
[Garay, Jakobsson, MacKenzie Crypto '99]



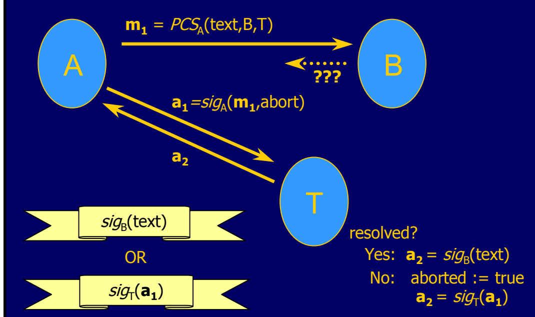
Role of Trusted Third Party

- T can convert PCS to regular signature
  - Resolve the protocol, when requested by either player
- T can issue an abort token
  - Promise not to resolve protocol in future
- T acts only when requested
  - Decides whether to abort or resolve on a first-come-first-served basis
  - Only gets involved if requested by A or B

Resolve Subprotocol



Abort Subprotocol



Fairness in ATL

$\neg\langle\langle B,Com\rangle\rangle\Diamond(contract_A \wedge \neg\langle\langle A_h\rangle\rangle\Diamond contract_B)$

Bob in collaboration with communication channels
does not have a strategy
to reach a state in which
Bob has Alice's signature
but honest Alice does not have a strategy
to reach a state in which Alice has Bob's signature

Timeliness + Fairness in ATL

$\langle\langle A_h\rangle\rangle\Diamond(stop_A \wedge (\neg contract_B \rightarrow \neg\langle\langle B,Com\rangle\rangle\Diamond contract_A))$

Honest Alice always has a strategy to reach a state
in which she can stop the protocol and
if she does not have Bob's contract
then Bob does not have a strategy to obtain
Alice's signature even if he controls
communication channels

Abuse-Freeness in ATL

$\neg\langle\langle A\rangle\rangle\Diamond(proveToC \wedge \langle\langle A\rangle\rangle\Diamond contract_B \wedge \langle\langle A\rangle\rangle\Diamond(aborted \wedge \neg\langle\langle B_h\rangle\rangle\Diamond contract_A))$

Alice doesn't have a strategy to reach state in which
she can prove to Charlie that
she has a strategy to obtain Bob's contract AND
a strategy to abort the protocol, i.e., reach a state where
Alice has received abort token and Bob doesn't have
a strategy to obtain Alice's signature

Modeling TTP and Communication

- Trusted third party is impartial
  - This is modeled by defining a unique TTP strategy
  - TTP has no choice: in every state, the next action is uniquely determined by its only strategy
- Can model protocol under different assumptions about communication channels
  - Unreliable: infinite delay possible, order not guaranteed
    - Add "idle" action to the channel state machine
  - Resilient: finite delays, order not guaranteed
    - Add "idle" action + special constraints to ensure that every message is eventually delivered (rule out infinite delay)
  - Operational: immediate transmission

MOCHA Model Checker

- Model checker specifically designed for verifying alternating transition systems
  - System behavior specified as guarded commands
    - Essentially the same as PRISM input, except that transitions are nondeterministic (as in in Murφ), not probabilistic
  - Property specified as ATL formula
- Slang scripting language
  - Makes writing protocol specifications easier
- Try online implementation!
  - http://www-cad.eecs.berkeley.edu/~mocha/trial/

Bibliography

- R. Alur, T. Henzinger, O. Kupferman. "Alternating-time temporal logic". FOCS '97.
  - Introduces alternating transition systems and ATL logic
- R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, S. Tasiran. "MOCHA: modularity in model checking". CAV '98.
  - Introduces MOCHA model checker for alternating transition systems
- http://www.cis.upenn.edu/~mocha/
  - MOCHA web page
- S. Kremer and J.-F. Raskin. "A game-based verification of non-repudiation and fair exchange protocols". J. of Computer Security 11(3), 2003.
  - Detailed study of fair exchange protocols using ATL and MOCHA
- S. Kremer and J.-F. Raskin. "Game-based analysis of abuse-free contract signing". CSFW '03.
  - Model checking abuse-freeness with MOCHA