Multiset Rewriting with Dense Times and the Analysis of Cyber-Physical Security Protocols

Max Kanovich, Tajana Ban Kirigin, Vivek Nigam, Andre Scedrov, and Carolyn Talcott

Cyber-Physical Security Protocols

Cyber-Physical Security Protocols are security protocols which rely **on the physical properties** in which its protocol sessions are carried out, such as:

- message transmission takes time;
- processing requests takes time;
- different transmission channels and velocities;
- physical and network distances between participants.

Cyber-Physical Security Protocols

Example: Distance Bounding Protocols

Cyber-Physical Security Protocols

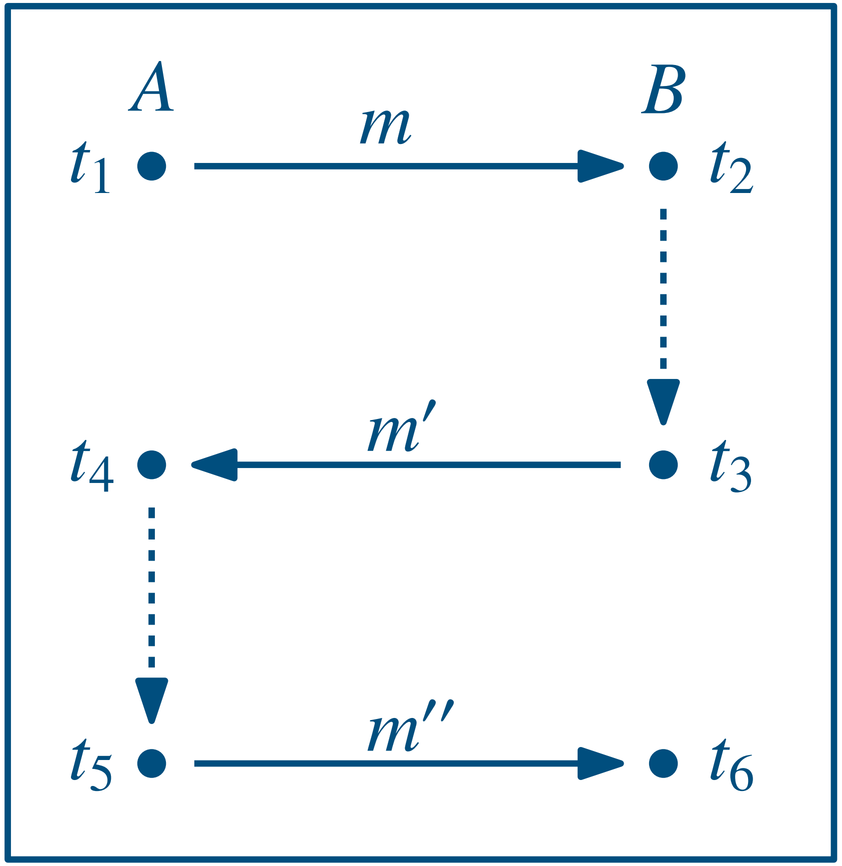
Example: Distance Bounding Protocols

The round trip time of messages and **the transmission velocity** is taken into account to **infer an upper bound of the distance** between two agents.

Cyber-Physical Security Protocols

Example: Distance Bounding Protocols

The round trip time of messages and **the transmission velocity** is taken into account to **infer an upper bound of the distance** between two agents.

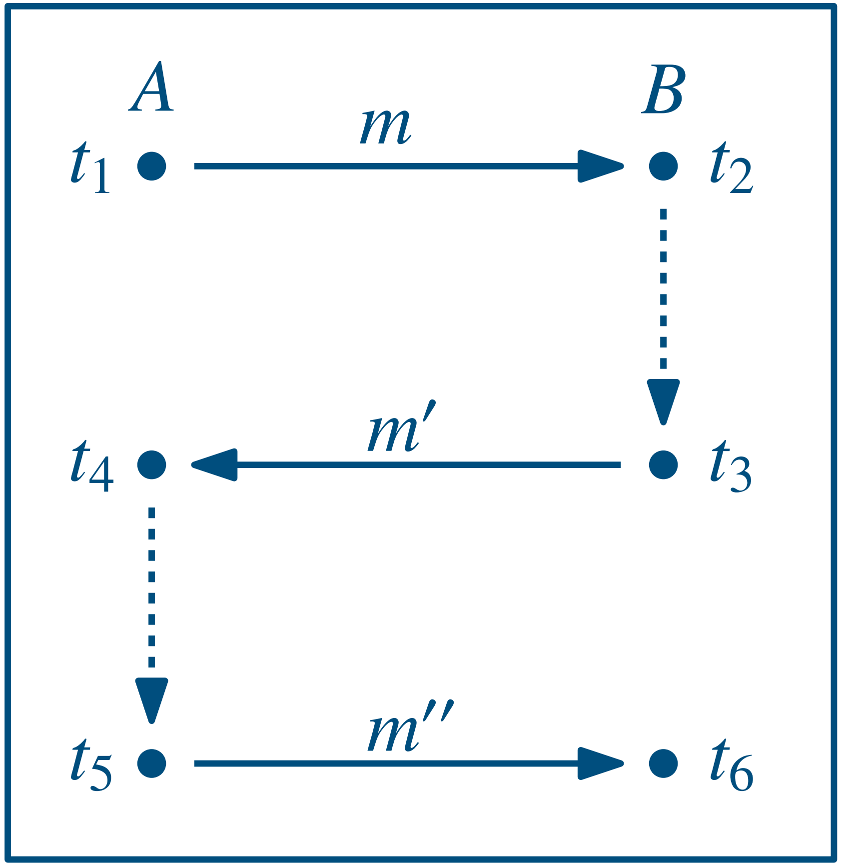


Cyber-Physical Security Protocols

Example: Distance Bounding Protocols

The round trip time of messages and **the transmission velocity** is taken into account to **infer an upper bound of the distance** between two agents.

If $t_4 - t_1 \leq R$ for a given **distance bounding time** R, then the verifier A grants the access to its resources to the prover B.



Specification of Cyber-Physical Security Protocols

Specification of Distance Bounding Protocols

Standard "Alice-Bob" notation needs to be refined.

$$
\begin{array}{ll}
A \longrightarrow B : m & \text{at time } t_0 \\
B \longrightarrow A : m' & \text{at time } t_1 \\
A \longrightarrow B : m'' & \text{if } t_1 - t_0 \leq R
\end{array}
$$

Specification of Cyber-Physical Security Protocols

Specification of Distance Bounding Protocols

Standard "Alice-Bob" notation needs to be refined.

$$
\begin{array}{ll}
A \longrightarrow B : m & \text{at time } t_0 \\
B \longrightarrow A : m' & \text{at time } t_1 \\
A \longrightarrow B : m'' & \text{if } t_1 - t_0 \leq R
\end{array}
$$

Many **assumptions about time** need to be formally specified, including:

- time requirements for the fulfillment of a protocol session;
- assumptions about the network, such as communication mediums and transmission velocities.

Protocol Analysis

Following issues need to be addressed:

- which properties does the protocol ensure;
- under which conditions;
- against which intruders.
- capabilities of the participants and the intruder(s) should be **ammended with time features** in order to make the physical properties of the system relevant.

Discrete vs Continuous Times

Discrete Time Models

Continuous Time Models

Discrete vs Continuous Times

Discrete Time Models

Continuous Time Models

We investigate how the models with continuous time relate to models with discrete time in protocol analysis.

In particular, we show that protocols proven secure in the discrete model **may be shown flawed** in the continuous model.

Agenda

Attack in Between Ticks

- MSR with Continuous Time
- Circle Configurations
- Conclusions and Future Work

Back to Distance Bounding Protocols

- Assume $R = 4$;
- Verifier needs to perform four operations:

  1) Send Challenge;

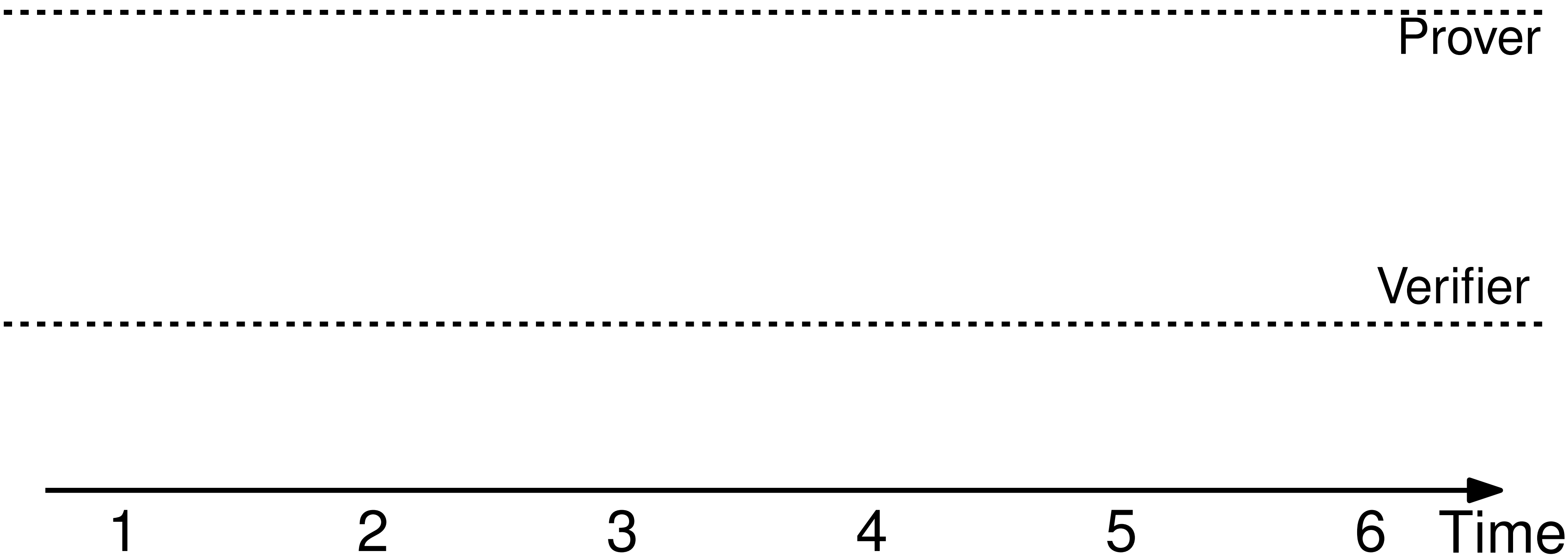
  2) Record time when message is sent;

  3) Receive Reponse;

  4) Record time when reponse is received.

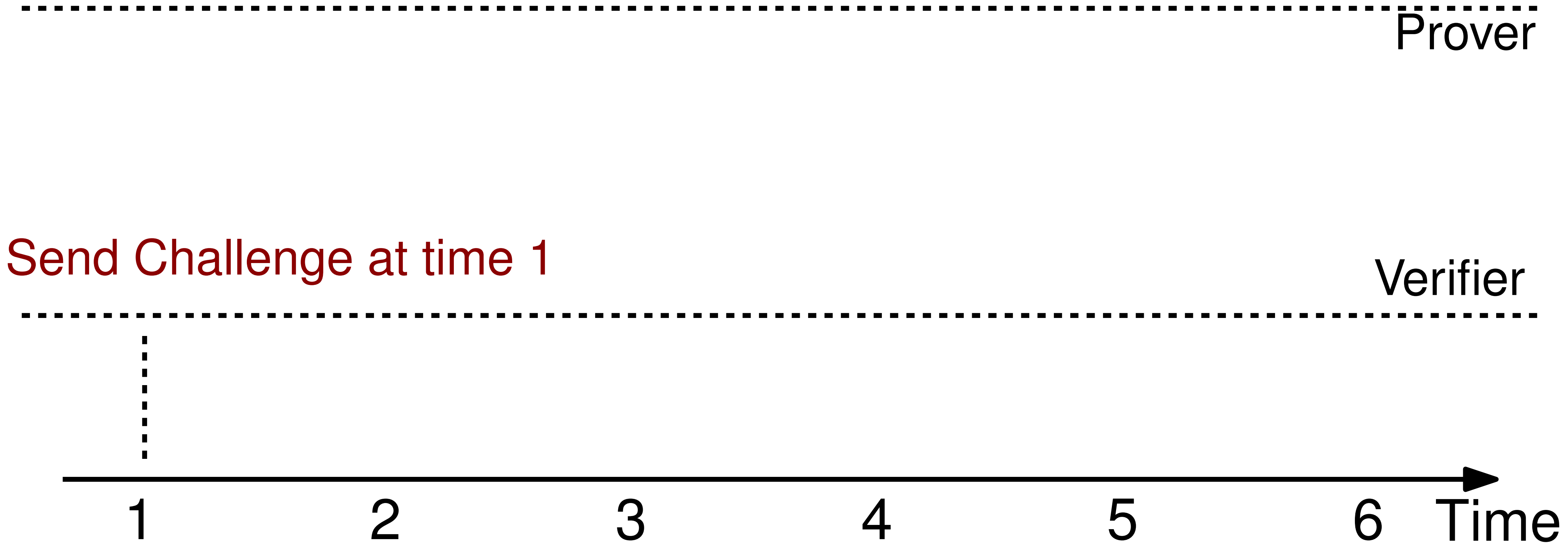
Back to Distance Bounding Protocols

Using a Discrete Model



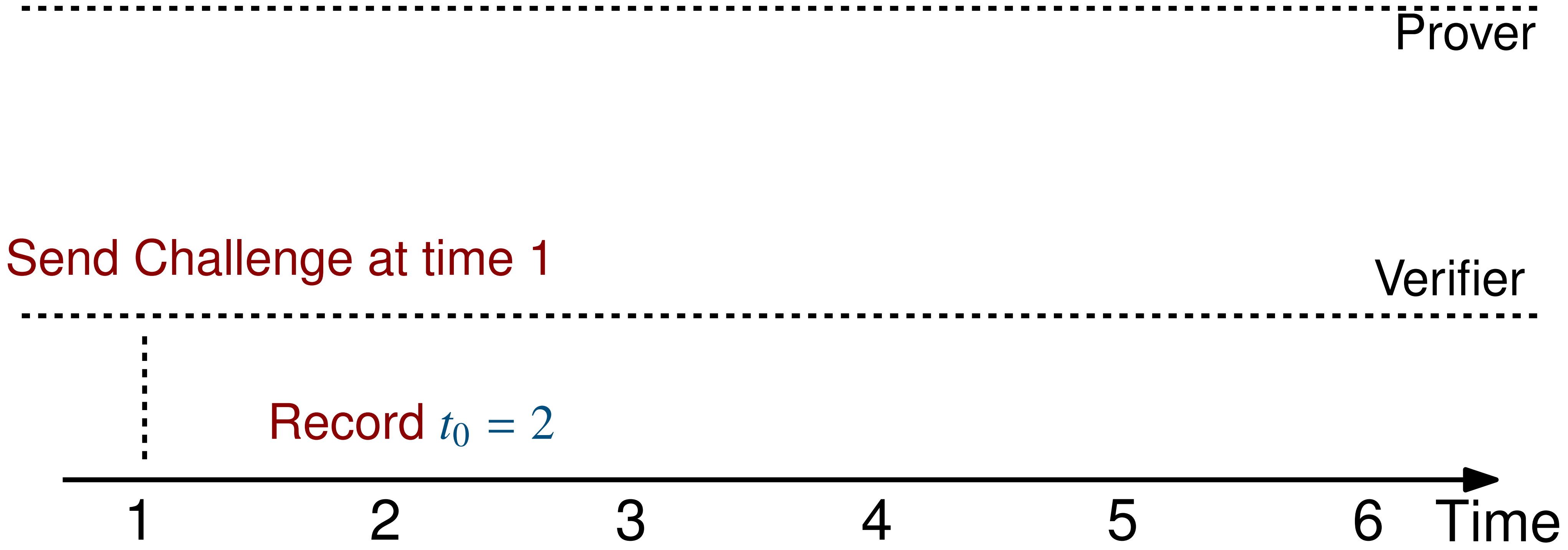
Back to Distance Bounding Protocols

Using a Discrete Model



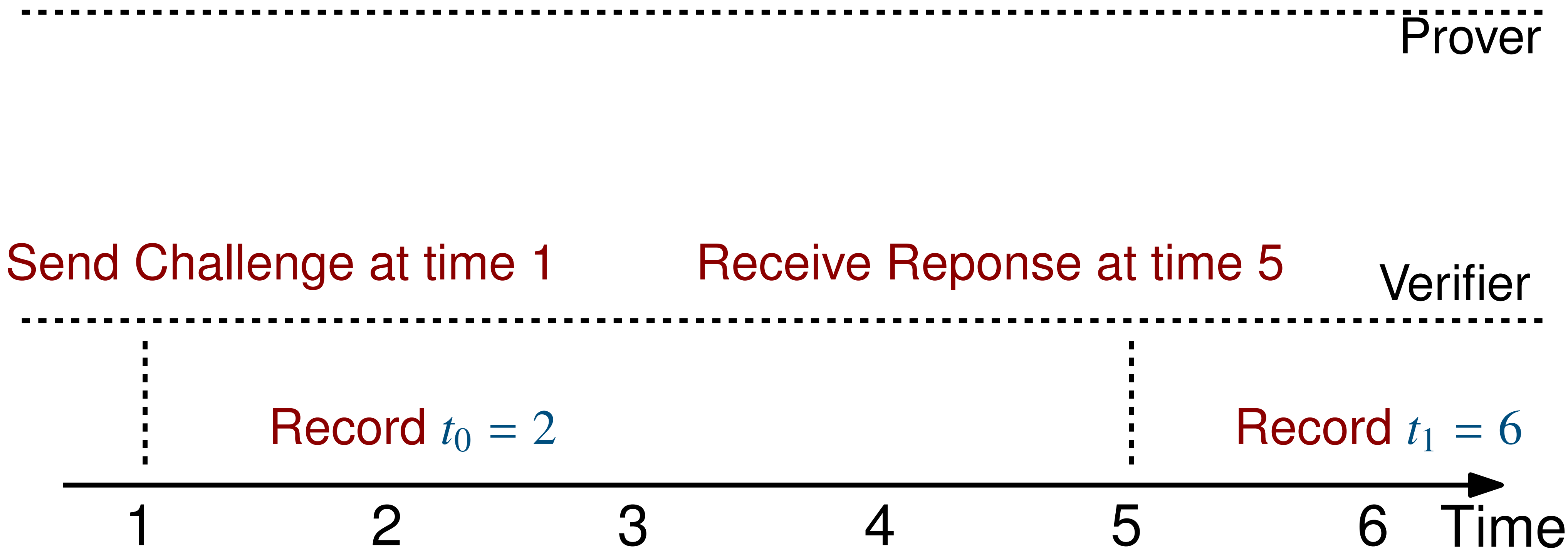
Back to Distance Bounding Protocols

Using a Discrete Model



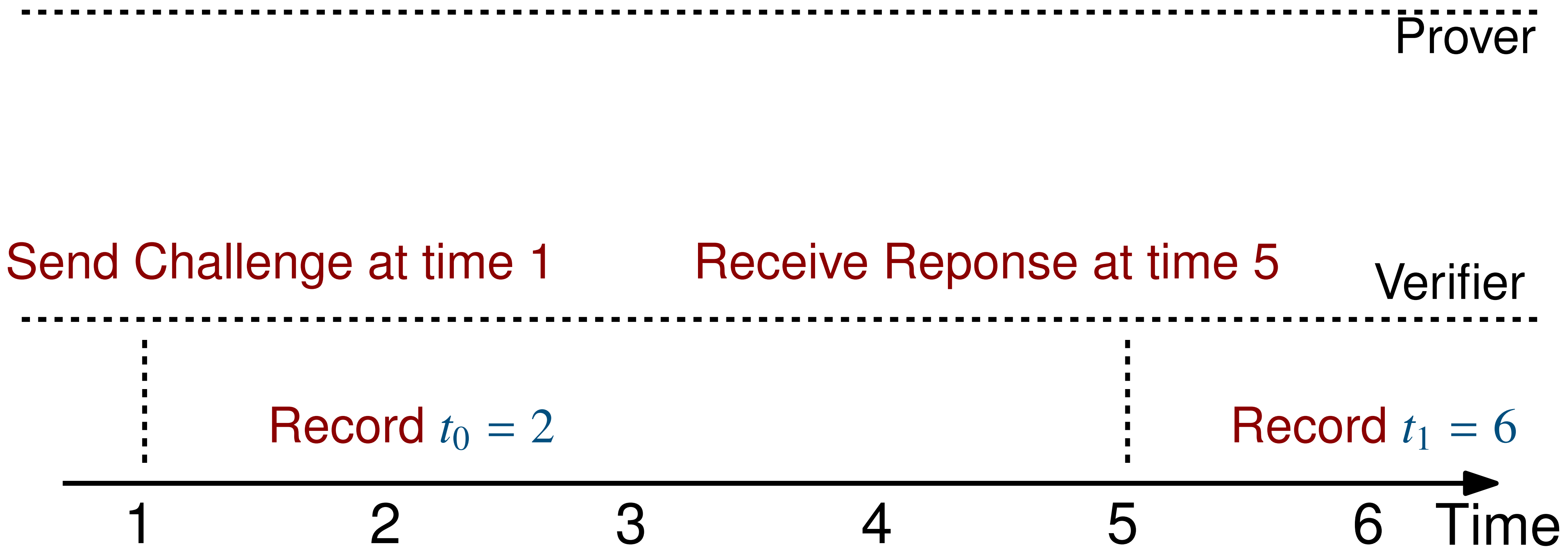
Back to Distance Bounding Protocols

Using a Discrete Model



Back to Distance Bounding Protocols

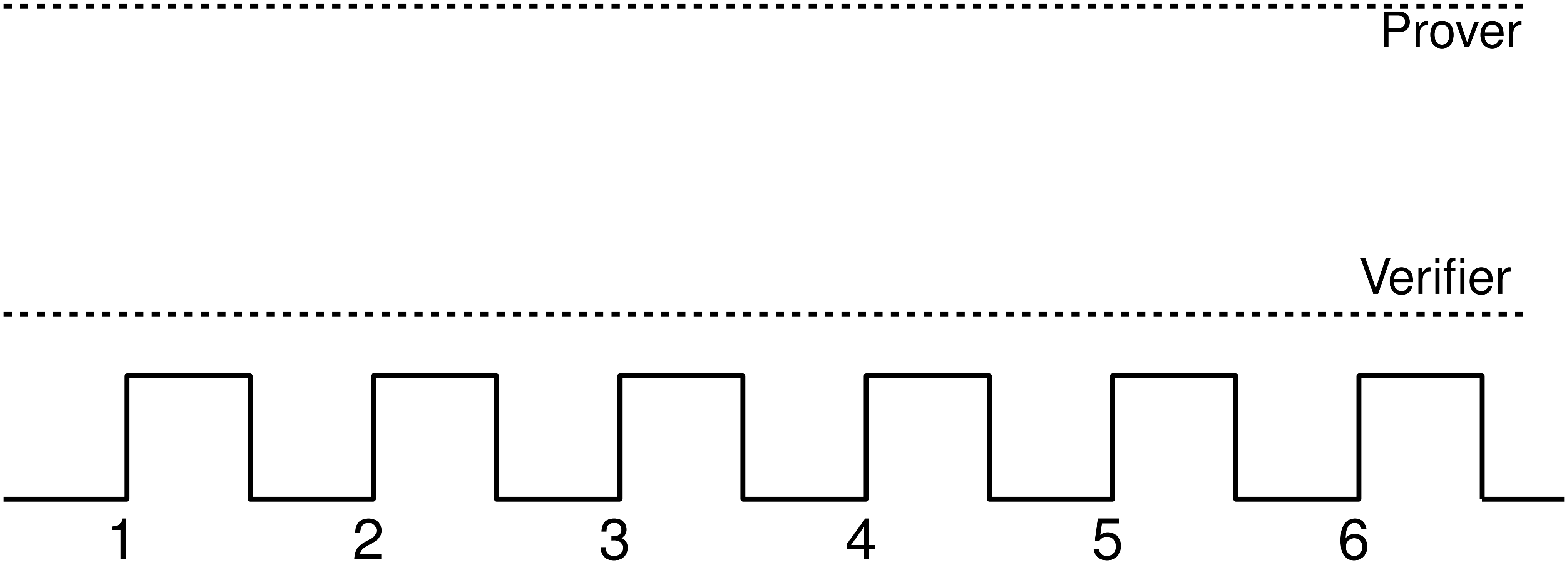
Using a Discrete Model



Verifier Grants access to the Prover as $t_1 - t_0 = 4$

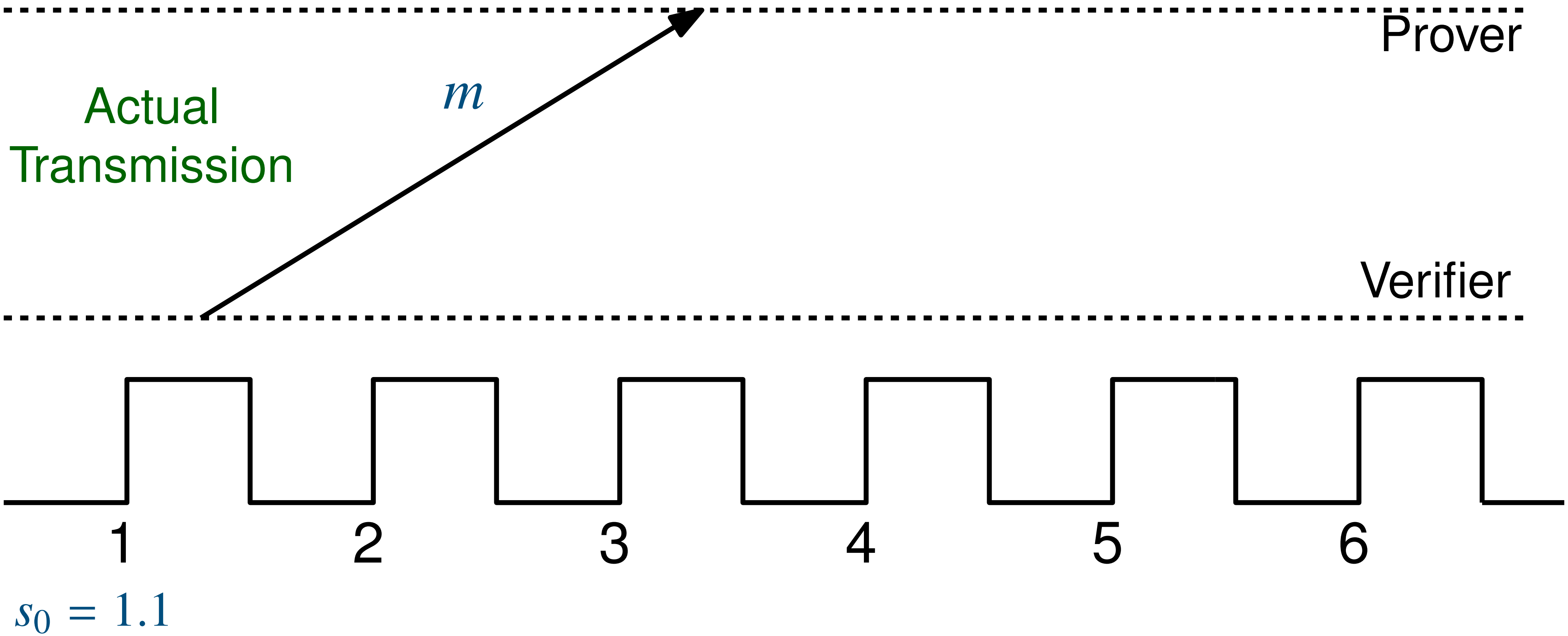
Attack in Between Ticks

Using a Continuous Model



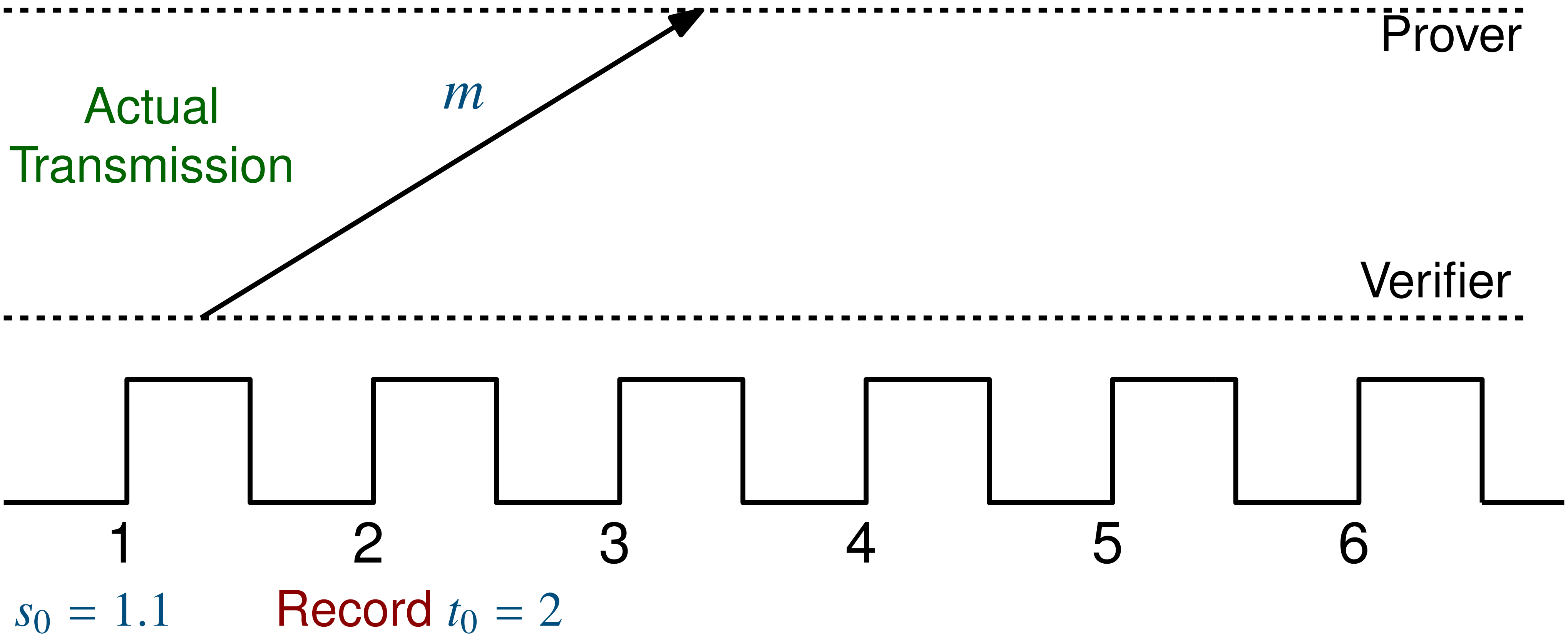
Attack in Between Ticks

Using a Continuous Model



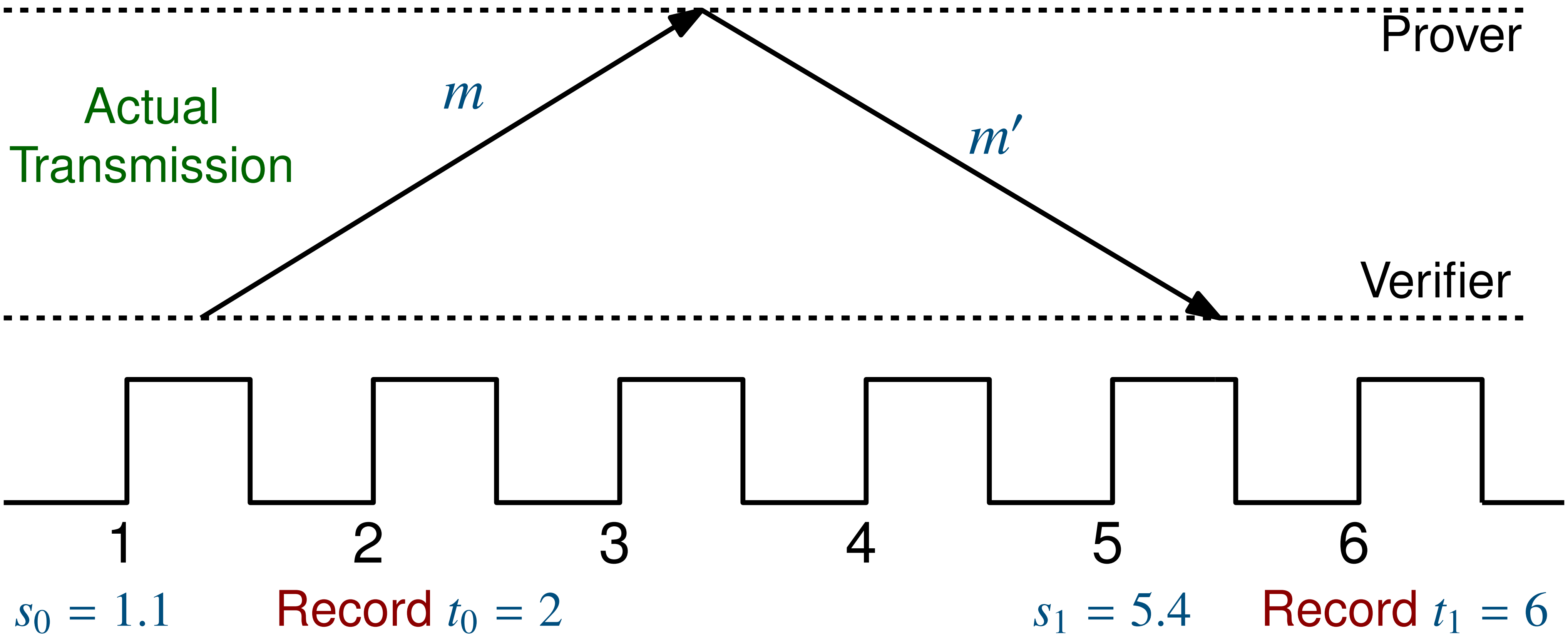
Attack in Between Ticks

Using a Continuous Model



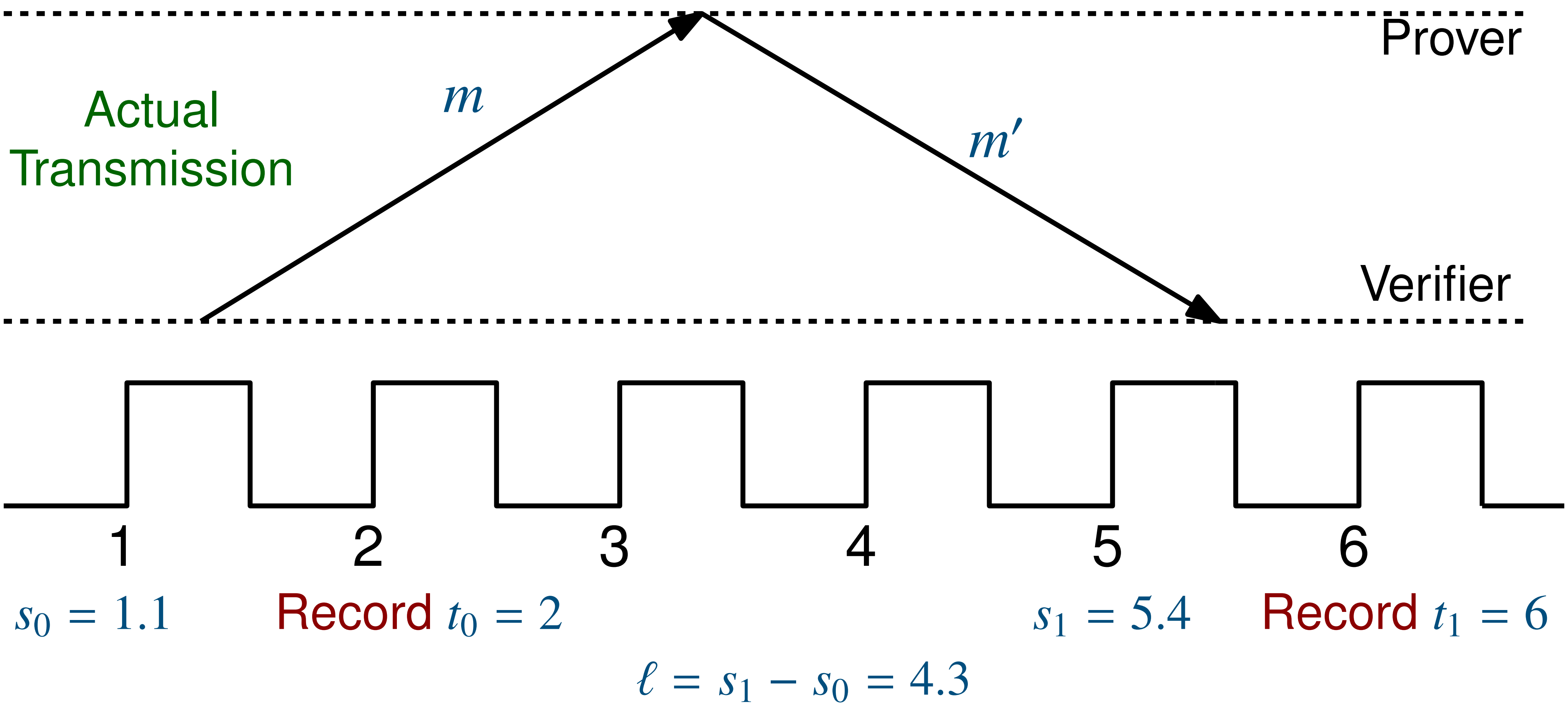
Attack in Between Ticks

Using a Continuous Model



Attack in Between Ticks

Using a Continuous Model



Verifier grants access, although actual round trip time is greater than R!

Back to Distance Bounding Protocols

The difference between actual round trip time and measured trip time can be of **one clock tick** even if each operation is executed in one clock cycle.

- 1 clock cycle of a 24MHz processor = 42 ns;
- Light travels 30cm in 1ns;
- Thus the error can be of 12.6 meters round trip, which means the prover can be **6.3 meters further** than the distance bound.

Agenda

- Attack in Between Ticks

MSR with Continuous Time

- Circle Configurations
- Conclusions and Future Work

Multiset Rewriting with Continuous Time

Inspired by the work [Okada, Kanovich and Scedrov ENTCS 98] and [DeYoung, Garg and Pfenning CSF 08].

Multiset Rewriting with Continuous Time

Inspired by the work [Okada, Kanovich and Scedrov ENTCS 98] and [DeYoung, Garg and Pfenning CSF 08].

- **Timestamped Facts:** – A Fact F with an associated real number t, written $F@t$;
- **Configuration** – A multiset of facts with **exactly one occurrence of Time**.

{Time@7.5, Deadline@10.3, Task(1,ok)@5.3, Task(2,todo)@2.13}

Multiset Rewriting with Continuous Time

- **Tick Rule** – Advances Global Time for any $\epsilon > 0$,

$$Time@T \longrightarrow Time@(T + \varepsilon)$$

Multiset Rewriting with Continuous Time

- **Tick Rule** – Advances Global Time for any $\epsilon > 0$,

$$Time@T \longrightarrow Time@(T + \varepsilon)$$

- **Instantaneous Rules** – Changes the state, but not the global time

$$Time@T, Task(1, ok)@T_1, Deadline@T_2, Task(2, todo)@T_3 \mid \{T_2 \geq T + 2\}$$
$$\longrightarrow Time@T, Task(1, ok)@T_1, Deadline@T_2, Task(2, ok)@(T + 1)$$

Multiset Rewriting with Continuous Time

- **Tick Rule** – Advances Global Time for any $\epsilon > 0$,

$$Time@T \longrightarrow Time@(T + \varepsilon)$$

- **Instantaneous Rules** – Changes the state, but not the global time

Time Constraints:
$T > T' \pm D$ and $T = T' \pm D$

$$Time@T, Task(1, ok)@T_1, Deadline@T_2, Task(2, todo)@T_3 \mid \{T_2 \geq T + 2\}$$
$$\longrightarrow Time@T, Task(1, ok)@T_1, Deadline@T_2, Task(2, ok)@(T + 1)$$

Timestamps of new facts: $T + D$

where the Ds are non-negative integers.

Multiset Rewriting with Continuous Time

Applying the rule:

$$\begin{array}{l} Time@T, Task(1,ok)@T_1, Deadline@T_2, Task(2,todo)@T_3 \mid \{T_2 \geq T+2\} \\ \quad \longrightarrow Time@T, Task(1,ok)@T_1, Deadline@T_2, Task(2,ok)@(T+1) \end{array}$$

to the configuration:

{Time@7.5, Deadline@10.3, Task(1,ok)@5.3, Task(2,todo)@2.13}

yields the configuration:

{Time@7.5, Deadline@10.3, Task(1,ok)@5.3, Task(2,ok)@8.5}

Multiset Rewriting with Continuous Time

- **Goal** – A pair of timestamped facts and time constraints.

$$S_G = \{F_1 @ T_1, \ldots, F_n @ T_n\} \mid C$$

where $T_1, \ldots, T_n$ are time variables, $F_1, \ldots, F_n$ are ground facts and C is a set of constraints involving only $T_1, \ldots, T_n$.

Multiset Rewriting with Continuous Time

- **Goal** – A pair of timestamped facts and time constraints.

$$\mathcal{S}_G = \{F_1@T_1, \ldots, F_n@T_n\} \mid C$$

where $T_1, \ldots, T_n$ are time variables, $F_1, \ldots, F_n$ are ground facts and C is a set of constraints involving only $T_1, \ldots, T_n$.

$\mathcal{S}_1$ is a **goal configuration** if there is a substitution σ such that:

- $\mathcal{S}_G\sigma \subseteq \mathcal{S}_1$
- all the constraints in $C\sigma$ are satisfied.

Multiset Rewriting with Continuous Time

- **Reachability Problem** – Given a set of actions and an initial configuration, **is there a goal configuration that can be reached** from the initial configuration using the given actions?

In our POST 2015 paper, we formalize the Attack in Between Ticks using our model.

Agenda

- Attack in Between Ticks
- MSR with Continuous Time
- **Circle Configurations**
- Conclusions and Future Work

Complexity Results

<table>
<tr><th colspan="3">Rechability Problem</th></tr>
<tr><td rowspan="3">Balanced Actions</td><td>Untimed System</td><td>PSPACE-complete
[Kanovich et al., IC’14]</td></tr>
<tr><td>Discrete Time</td><td>PSPACE-complete
[Kanovich et al., MSCS’15]</td></tr>
<tr><td>System with dense time</td><td>PSPACE-complete new</td></tr>
<tr><td colspan="2">Actions not necessarily balanced</td><td>Undecidable</td></tr>
</table>

The PSPACE-completeness results also hold for systems that can create an **unbounded number of fresh values**, such as nonces.

Challenge

We need to handle the non-determinism caused by the tick rule:

$$Time@T \longrightarrow Time@(T + \varepsilon)$$

Here $\varepsilon > 0$ can be any positive real number. So how to advance time?

Solution

We propose a new **equivalence class** on configurations.

Circle Configurations

Solution by Example

Consider a system $\mathcal{T}$ and assume that the greatest natural number, D_{max} in $\mathcal{T}$ is 3.

Consider the following configuration:

$$S_1 = \{P_0@0.4, P_1@1.5, Time@5.4, P_2@7.6\}$$

Its circle configuration is composed of **two parts**:

Solution by Example

Consider a system $\mathcal{T}$ and assume that the greatest natural number, D_{max} in $\mathcal{T}$ is 3.

Consider the following configuration:

$$S_1 = \{P_0@0.4, P_1@1.5, Time@5.4, P_2@7.6\}$$

Its circle configuration is composed of **two parts**:

- δ**-configuration** – constructed using time differences of the integer part of timestamps truncated by D_{max}.

$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$

Solution by Example

Consider a system $\mathcal{T}$ and assume that the greatest natural number, D_{max} in $\mathcal{T}$ is 3.

Consider the following configuration:

$$S_1 = \{P_0@0.4, P_1@1.5, Time@5.4, P_2@7.6\}$$

Its circle configuration is composed of **two parts**:

- δ**-configuration** – constructed using time differences of the integer part of timestamps truncated by D_{max}.

$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$

$$\lfloor 1.5 \rfloor - \lfloor 0.4 \rfloor \qquad \lfloor 5.4 \rfloor - \lfloor 1.5 \rfloor > D_{max} \qquad \lfloor 7.6 \rfloor - \lfloor 5.4 \rfloor$$

Solution by Example

Consider the following configuration:

$$S_1 = \{P_0@0.4, P_1@1.5, Time@5.4, P_2@7.6\}$$

Its circle configuration is composed of **two parts**:

- **unit-configuration** – order the facts according to **the decimal part** of their timestamps.

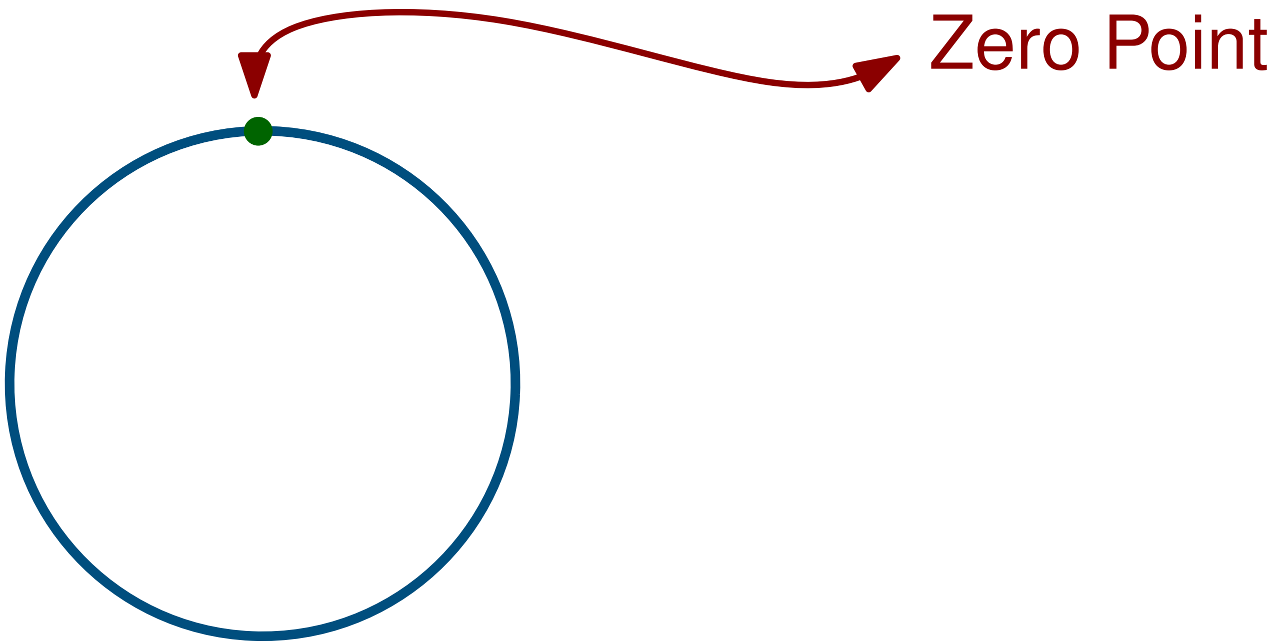
Solution by Example

Consider the following configuration:

$$S_1 = \{P_0@0.4, P_1@1.5, Time@5.4, P_2@7.6\}$$

Its circle configuration is composed of **two parts**:

- **unit-configuration** – order the facts according to **the decimal part** of their timestamps.



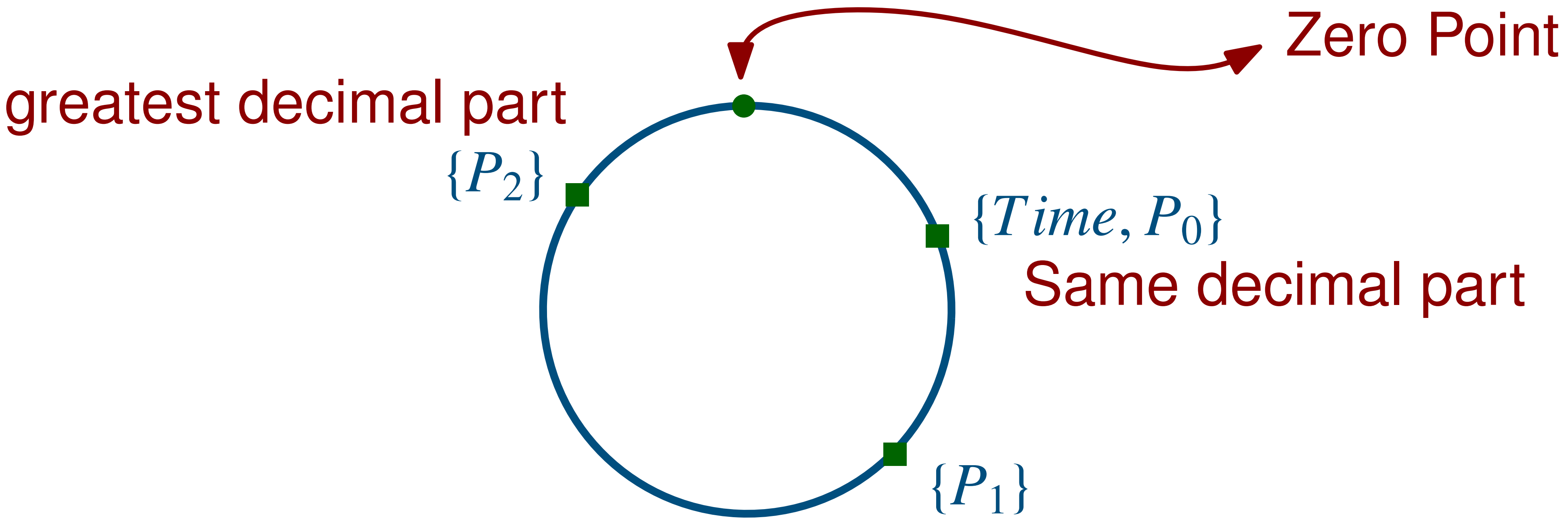
Solution by Example

Consider the following configuration:

$$S_1 = \{P_0@0.4, P_1@1.5, Time@5.4, P_2@7.6\}$$

Its circle configuration is composed of **two parts**:

- **unit-configuration** – order the facts according to **the decimal part** of their timestamps.



Solution by Example

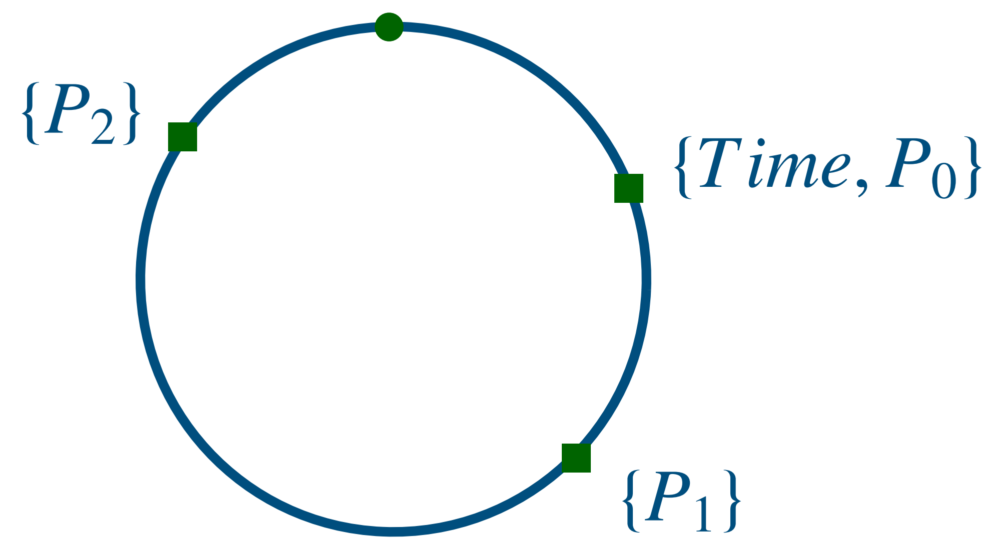
The following configurations are **equivalent**:

$$S_1 = \{P_0@0.4, P_1@1.5, Time@5.4, P_2@7.6\}$$

$$S_2 = \{P_0@3.2, P_1@4.4, Time@9.2, P_2@11.7\}$$

because they have the **same circle configuration**:

$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$



Executable Model with Circle Configuration

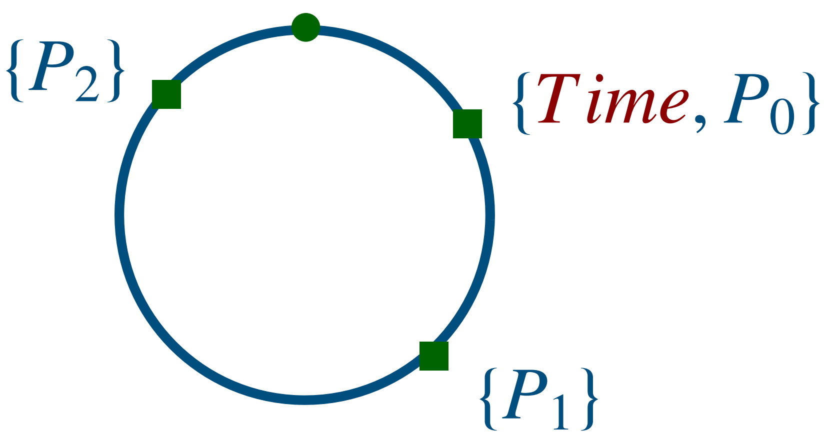
We can execute actions **on circle configurations** instead of **concrete configurations**:

Executable Model with Circle Configuration

We can execute actions **on circle configurations** instead of **concrete configurations**:

One case for the Tick Rule

$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$

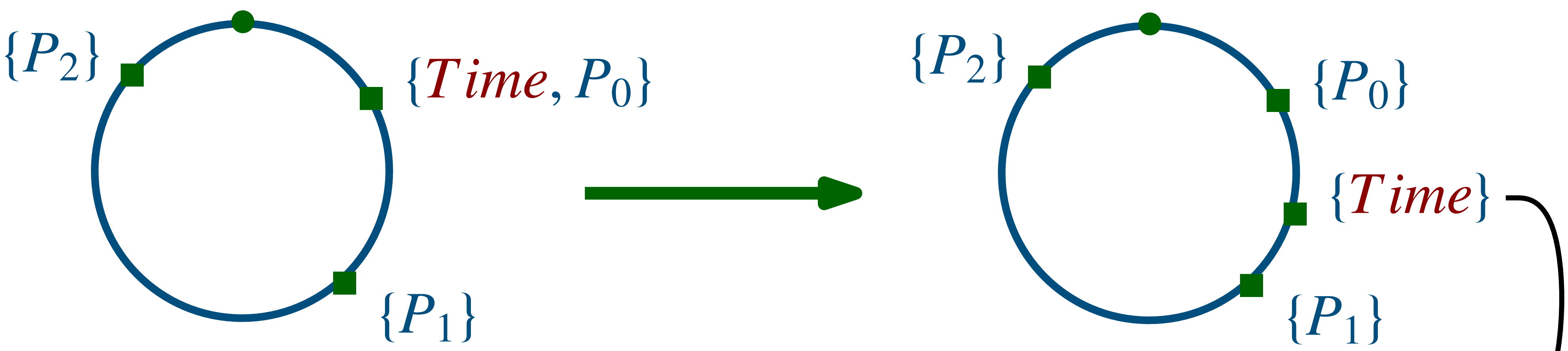


Executable Model with Circle Configuration

We can execute actions **on circle configurations** instead of **concrete configurations**:

One case for the Tick Rule

$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$

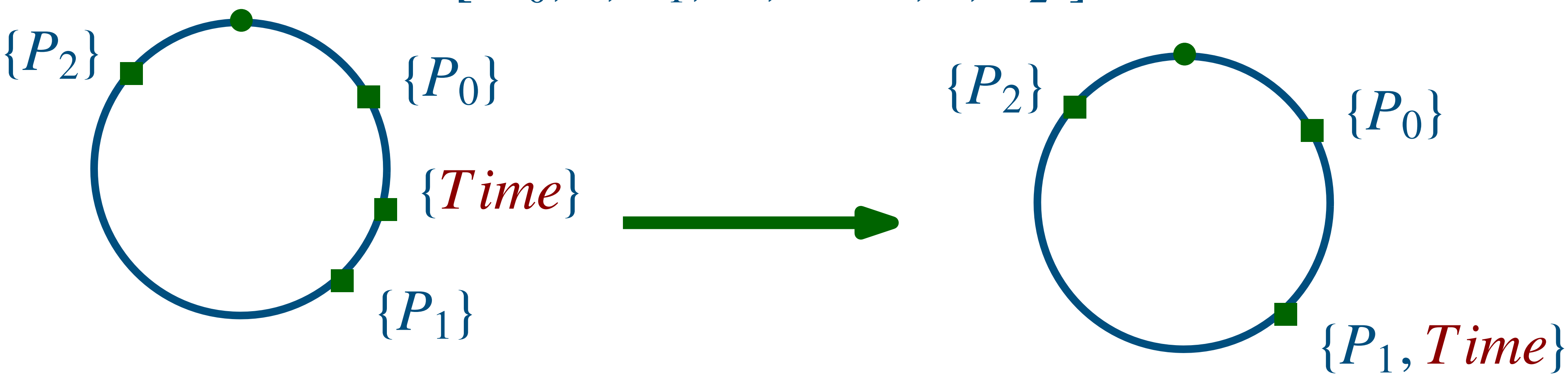


Time shifting **by any** ϵ so that the decimal part of Time is between the decimal part of P_0 and P_1.

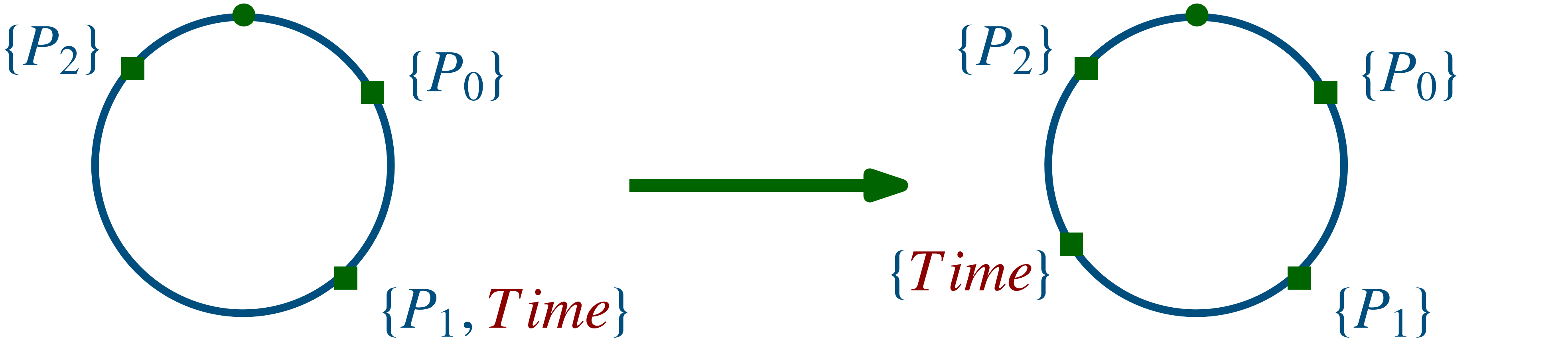
Executable Model with Circle Configuration

Other cases for the Tick Rule

$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$



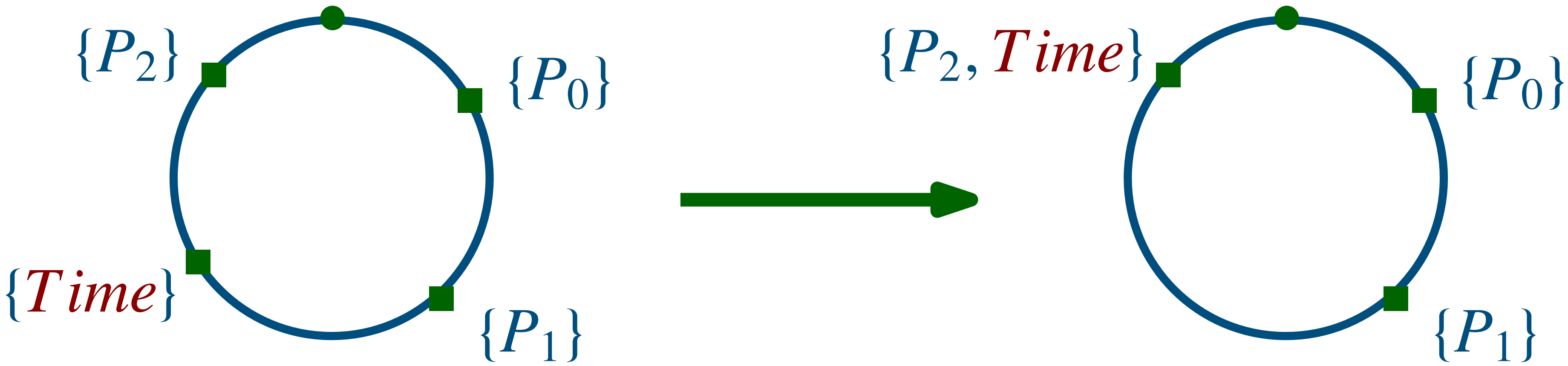
$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$



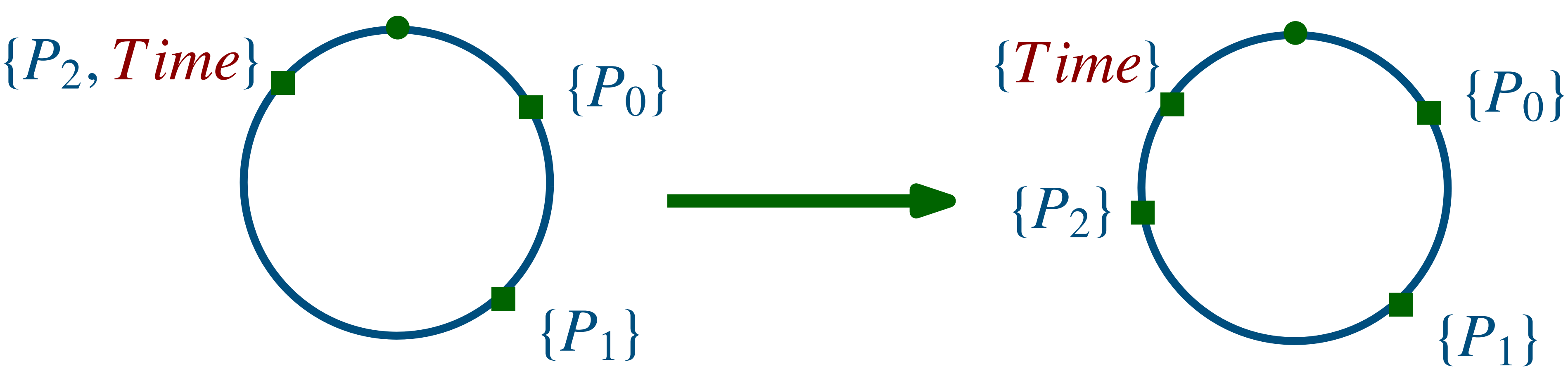
Executable Model with Circle Configuration

Other cases for the Tick Rule

$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$



$$[\ P_0, 1, P_1, \infty, Time, 2, P_2\ ]$$

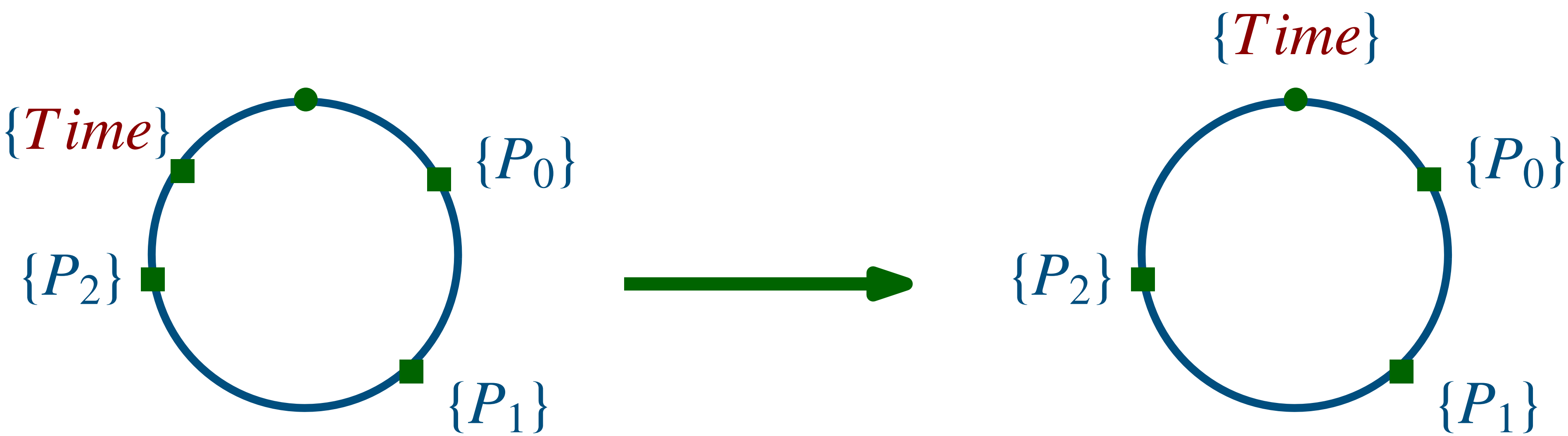


Executable Model with Circle Configuration

Other cases for the Tick Rule

When $Time$ completes a round on the circle, the integer differences are updated.

$[\ P_0, 1, P_1, \infty, Time, \mathbf{2}, P_2\ ]$ $\qquad$ $[\ P_0, 1, P_1, \infty, Time, \mathbf{1}, P_2\ ]$



Results

Lemma: Let $\mathcal{T}$ be a reachability problem and D_{max} be an upper bound on the numeric values in $\mathcal{T}$. Let $\mathcal{S}_1$ be a configuration, whose circle-configuration is C_1, and r be an instantaneous action in $\mathcal{T}$. Then $\mathcal{S}_1 \longrightarrow_r \mathcal{S}_2$ if and only if $C_1 \longrightarrow_{[r]} C_2$ and C_2 is the circle-configuration of $\mathcal{S}_2$. Moreover, $\mathcal{S}_1 \longrightarrow_{Tick} \mathcal{S}_2$ if and only if $C_1 \longrightarrow^*_{Next} C_2$ and C_2 is the circle-configuration of $\mathcal{S}_2$.

Results

Theorem: Let $\mathcal{T}$ be a reachability problem, D_{max} be an upper bound on the numeric values in $\mathcal{T}$. Then $\mathcal{S}_I \longrightarrow^* \mathcal{S}_G$ for some initial and goal configurations, $\mathcal{S}_I$ and $\mathcal{S}_G$, in $\mathcal{T}$ if and only if $C_I \longrightarrow^* C_G$ where C_I and C_G are the circle-configurationss of $\mathcal{S}_I$ and $\mathcal{S}_G$, respectively.

Agenda

- Attack in Between Ticks
- MSR with Continuous Time
- Circle Configurations
- **Conclusions and Future Work**

Conclusions

- We investigated the impacts on analysis of protocols when using models with discrete time and with dense times;
- We discovered a novel attack on Distance Bounding Protocols called Attack in Between Ticks;
- We proposed a model with continuous time based on multiset rewriting;
- We proved that the reachability problem for balanced timed systems is PSPACE-complete.

Related Work

- Timed Automata [Alur 1994]:
  - – Our Attack-Between Ticks is inspired by similar issues regarding discrete vs. dense time in the analysis of digital circuits
  - – Both our PSPACE complexity results and our proof method are quite different from timed automata

- Analysis and verification of Distance Bounding Protocols
  - – Meadows, Pavlovic et al. [2007, 2009]
  - – Verification in Isabelle [Basin et al. 2011]

Future Work

- Implementation in Maude with SMT of our systems. We already implemented the machinery which checks whether a systems is vulnerable to the Attack in Between Ticks;

- Investigate ways to mitigate the Attack in Between Ticks: for example, examine the impacts of using several challenge-response rounds;

- Formalize other anomalies, such as those involving privacy violations using RFID passports.

Question?

Thank You!

Related Work

- Timed Automata are different to our formalism (see our MSCS 15 paper for a detailed comparison):
  - – TA does not support nonces nor quantification;
  - – PSPACE-completeness proofs are different: we do not use regions.
- Basin et al.'s work on Distance Bounding Verification:
  - – Their Isabelle formalization considered the prover running in dense time (no clocks);
  - – Attack in Between the Clicks was not foreseen by their model;
  - – No complexity results.
- Meadows et al.'s work on Distance Bounding Verification:
  - – Also did not consider provers using clocks and therefore did not foresee the Attack in Between Ticks;
  - – No executable model nor complexity results.