Distance Bounding Protocols: Computational vs. Symbolic Models

Jorge Toro Pozo
University of Luxembourg

(joint work with S. Mauw, Z. Smith and R. Trujillo)

FutureDB Workshop
Azores, Portugal - April 14, 2018
Outline

1. Introduction
2. Probabilistic model based on automata
3. Symbolic model with time and location
4. Symbolic model based on causality
5. Conclusion and Future


Problem: Relay attack

A relay attack is a man-in-the-middle attack in which an attacker relays verbatim a message from the sender to a valid receiver.

Source: securepositioning.com
Problem: Relay attack

A relay attack is a man-in-the-middle attack in which an attacker relays verbatim a message from the sender to a valid receiver.

Source: securepositioning.com
Solution: Distance-bounding protocols

**Definition**

A *distance-bounding protocol* is an authentication protocol that checks that the distance between verifier and prover is below a given threshold.
Definition

A *distance-bounding protocol* is an authentication protocol that checks that the distance between verifier and prover is below a given threshold.

How to measure (or bound) distance?

- Verifier sends a challenge.
- Prover provides corresponding response.
- Verifier measures the round-trip-time.
A challenge/response round

\[ \text{dist}(V, P) = \frac{1}{2} \cdot c \cdot (\Delta t - t_{\text{comp}}) \leq \frac{1}{2} \cdot c \cdot \Delta t \]
Outline

1. Introduction
2. Probabilistic model based on automata
3. Symbolic model with time and location
4. Symbolic model based on causality
5. Conclusion and Future
To obtain an accurate upper-bound on the distance, the computational time on the prover’s side must be as short as possible.

Solution: Pre-computing the possible responses and store them in a constant-time-access structure, such as a lookup-table.

Protocols with a final crypto-verification phase could be outperformed by a precomputation-based protocol with more rounds, with no increase of the computational cost: \[ \forall n, \exists m: \left(\frac{1}{2}\right)^n > \left(\frac{3}{4}\right)^m. \]

Partial information can be given if the protocol gets interrupted before finishing.
Lookup protocols are DB protocols such that:

1. In the fast phase, the responses to the challenges are the result of lookup operations from a table.
Lookup protocols are DB protocols such that:

1. In the fast phase, the responses to the challenges are the result of lookup operations from a table.

2. either do **NOT** have a final verification phase at all or
Lookup protocols are DB protocols such that:

1. In the fast phase, the responses to the challenges are the result of lookup operations from a table.

2. either do NOT have a final verification phase at all or

3. having replied correctly and on time to all challenges is SUFFICIENT to pass the protocol (do not have any crypto-based verification mechanism such as opening commits, keyed hash functions, signatures...).
A = (Σ, Γ, Q, q₀, δ, ℓ)

Σ is the set of input symbols
Γ is the set of output symbols
Q is the set of states
q₀ ∈ Q is the initial state
δ: Q × Σ → Q is the transition function
ℓ: Q → Γ is the state labeling function
A = (Σ, Γ, Q, q₀, δ, ℓ)

- Σ is the set of input symbols
- Γ is the set of output symbols
- Q is the set of states
- q₀ ∈ Q is the initial state
- δ: Q × Σ → Q is the transition function
- ℓ: Q → Γ is the state labeling function

Ωₐ (101) = 001
$P = \{ q_0 \}
\begin{array}{cccc}
1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 \\
\end{array}
\ldots$
Protocol Execution

Slow/Lazy/Initial phase

Reader

Tag

Jorge Toro Pozo (Univ. of Luxembourg)
Distance-bounding protocols
April 15, 2018 14 / 45
Protocol Execution

Slow/Lazy/Initial phase

Distance-bounding phase

Reader

Tag

\[ \Delta t \leq t_{\text{MAX}} \]
Protocol Execution

Slow/Lazy/Initial phase

Distance-bounding phase

\[ \Delta t \leq t_{\text{MAX}} \]

Reader

Tag
Protocol Execution

Slow/Lazy/Initial phase

Distance-bounding phase

Reader

Tag

\[ \Delta t \leq t_{MAX} \]

\[ \Delta t \leq t_{MAX} \]

\[ \Delta t \leq t_{MAX} \]

\[ \Delta t \leq t_{MAX} \]
• State-label-insensitive relation $(\sim_s)$

$$(\Sigma, \Gamma, Q, q_0, \delta, \ell) \sim_s (\Sigma, \Gamma, Q, q_0, \delta, \ell')$$
Label-insensitive relation ($\sim_L$)

$$(\Sigma, \Gamma, Q, q_0, \delta, \ell) \sim_L (\Sigma, \Gamma, Q, q_0, \delta', \ell')$$

such that $\forall q \in Q : \{\delta(q, c) | c \in \Sigma\} = \{\delta'(q, c) | c \in \Sigma\}$. 

![Diagram showing automata equivalence relations](image-url)
A protocol $P$ is consistent w.r.t $\sim_R$ iff

$$A, A' \in P: A \sim_R A'$$
A protocol $P$ is consistent w.r.t $\sim_R$ iff

$$A, A' \in P: A \sim_R A'$$

A protocol $P$ is closed under $\sim_R$ iff

$$\forall (A, A') \in \sim_R: A \in P \implies A' \in P$$
Consistency and Closeness

- A protocol $P$ is consistent w.r.t $\sim_R$ iff
  \[ A, A' \in P : A \sim_R A' \]

- A protocol $P$ is closed under $\sim_R$ iff
  \[ \forall (A, A') \in \sim_R : A \in P \implies A' \in P \]

- The closure of $P$ w.r.t $\sim_R$, denoted by $P^R$, is the minimal superset of $P$ that is closed under $\sim_R$. 
Some formulas

Given a layered automaton $A$: 

$$ \text{mafia} \left( \{A\}^S \right) = \frac{1}{|\Sigma|^n \cdot |\Gamma|^n} \max_{x \in \Sigma^n} \left\{ \sum_{y \in \Sigma^n} |\Gamma|^{\text{collisions}_A(x,y)} \right\} $$

$$ \text{mafia} \left( \{A\}^L \right) = \frac{1}{|\Sigma|^{2n} \cdot |\Gamma|^n} \sum_{x, y \in \Sigma^n} |\Gamma|^{\text{collisions}_A(x,y)} $$
Some formulas

Given a layered automaton $A$:

$$mafia \left( \{ A \}^S \right) = \frac{1}{|\Sigma|^n \cdot |\Gamma|^n} \max_{x \in \Sigma^n} \left\{ \sum_{y \in \Sigma^n} |\Gamma|^{\text{collisions}_A(x,y)} \right\}$$

$$mafia \left( \{ A \}^L \right) = \frac{1}{|\Sigma|^{2n} \cdot |\Gamma|^n} \sum_{x, y \in \Sigma^n} |\Gamma|^{\text{collisions}_A(x,y)}$$

Trivially, $mafia \left( \{ A \}^S \right) \geq mafia \left( \{ A \}^L \right)$

because $\max \{ u_i \} \geq \frac{1}{N} (u_1 + \cdots + u_N)$
For any layered lookup protocol \( P \) the following holds:

\[
\text{mafia}(P) \geq \text{mafia}(P^S) \geq \\
\text{mafia}\left(\{A\}^L\right) \geq \text{mafia}(\text{Tree}),
\]

for some \( A \in P \).
Main theorem

**Theorem**

For any layered lookup protocol $P$ the following holds:

$$\text{mafia}(P) \geq \text{mafia}\left(P^S\right) \geq \text{mafia}\left(\{A\}^L\right) \geq \text{mafia}\left(\{M_{\text{size}}(P)\}^L\right) \geq \text{mafia}(\text{Tree}),$$

for some $A \in P$. 
Conclusions

- We have formalized relevant structural properties of lookup protocols that have been used in a rather intuitive way.
- We provided simple formulas for computing mafia success probability for all but one lookup protocols.
- We have addressed (partially) the security-memory trade-off problem in layered protocols.
Outline

1. Introduction

2. Probabilistic model based on automata

3. **Symbolic model with time and location**

4. Symbolic model based on causality

5. Conclusion and Future
The basis of our work
Model based on time and location


**Agents**: the set $Agent$, partitioned into $\{Honest, Dishonest\}$. 
Specification and Execution

- **Agents**: the set $Agent$, partitioned into $\{Honest, Dishonest\}$.
- **Messages**: the set $Msg$ defined by:

$$ m ::= atom \mid (m, m') \mid f(m) \mid \{m\}_{m'} $$

where $atom \in Nonce \cup Agent \cup Const$ and $f \in Fun$. 
Specification and Execution

- **Agents**: the set $Agent$, partitioned into $\{Honest, Dishonest\}$.
- **Messages**: the set $Msg$ defined by:

\[ m ::= \text{atom} \mid (m, m') \mid f(m) \mid \{m\}_{m'} \]

where $atom \in Nonce \cup Agent \cup Const$ and $f \in Fun$.
- **Events**: the set $Ev$ defined by:

\[ e ::= send_A(m) \mid recv_A(m) \mid claim_A(B, e', e'') \]
Specification and Execution

- **Agents**: the set $\text{Agent}$, partitioned into $\{\text{Honest}, \text{Dishonest}\}$.
- **Messages**: the set $\text{Msg}$ defined by:
  
  $$m ::= \text{atom} \mid (m, m') \mid f(m) \mid \{m\}_{m'}$$

  where $\text{atom} \in \text{Nonce} \cup \text{Agent} \cup \text{Const}$ and $f \in \text{Fun}$.
- **Events**: the set $\text{Ev}$ defined by:
  
  $$e ::= \text{send}_A(m) \mid \text{recv}_A(m) \mid \text{claim}_A(B, e', e'')$$

- **Trace**: a sequence $(t_1, e_1) \cdots (t_n, e_n)$ with $t_i \in \mathbb{R}, e_i \in \text{Ev}$.
**Agents:** the set \( \text{Agent} \), partitioned into \( \{ \text{Honest}, \text{Dishonest} \} \).

**Messages:** the set \( \text{Msg} \) defined by:

\[
m ::= \text{atom} \mid (m, m') \mid f(m) \mid \{m\}_{m'}
\]

where \( \text{atom} \in \text{Nonce} \cup \text{Agent} \cup \text{Const} \) and \( f \in \text{Fun} \).

**Events:** the set \( \text{Ev} \) defined by:

\[
e ::= \text{send}_A(m) \mid \text{recv}_A(m) \mid \text{claim}_A(B, e', e'')
\]

**Trace:** a sequence \((t_1, e_1) \cdots (t_n, e_n)\) with \( t_i \in \mathbb{R}, e_i \in \text{Ev} \).

**Specification:** a set of rules defining the actions of honest agents.
Agents: the set Agent, partitioned into \{Honest, Dishonest\}.

Messages: the set \( \text{Msg} \) defined by:

\[
m ::= \text{atom} \mid (m, m') \mid f(m) \mid \{m\}_{m'}
\]

where \( \text{atom} \in \text{Nonce} \cup \text{Agent} \cup \text{Const} \) and \( f \in \text{Fun} \).

Events: the set \( \text{Ev} \) defined by:

\[
e ::= \text{send}_A(m) \mid \text{recv}_A(m) \mid \text{claim}_A(B, e', e'')
\]

Trace: a sequence \((t_1, e_1) \cdots (t_n, e_n)\) with \( t_i \in \mathbb{R}, e_i \in \text{Ev} \).

Specification: a set of rules defining the actions of honest agents.

And some other stuff such as message deduction.
**Trace**: a sequence \((t_1, e_1) \cdots (t_n, e_n)\) with \(t_i \in \mathbb{R}, e_i \in Ev\).

\[
\alpha = (1.3, \text{send}_{Alice}(m)) \cdot (3, \text{recv}_{Bob}(m)) \cdot (5, \text{send}_{Bob}(h(m)))
\]
Specification and Execution

- **Trace**: a sequence \((t_1, e_1) \cdots (t_n, e_n)\) with \(t_i \in \mathbb{R}, e_i \in Ev\).

\[
\alpha = (1.3, send_{Alice}(m)) \cdot (3, recv_{Bob}(m)) \cdot (5, send_{Bob}(h(m)))
\]

\[
dist(Alice, Bob) \leq c \cdot (3 - 1.3)
\]
- **Specification**: a set of rules defining the actions of *honest* agents.

\[ \mathcal{P} = \{ R_1 \ldots, R_n \} \]  
where the \( R_i \)'s have the form:

\[
t \geq \max \{ \alpha \} \quad A \in \text{Honest} \\
\text{cond}_1 \quad \cdots \quad \text{cond}_n \\
(\alpha, (t, e)) \in R_i
\]

In words: if conditions \textit{cond}_j are met, then the agent \( A \) can execute the event \( e \) at time \( t \).
Example
Hancke and Kuhn’s 2005

\[ P = \{ R_1, R_2, R_3, R_4, R_5 \} \]

\[ V \in \text{Hnst} \]

\[ t \geq \max t(\alpha) \]

\[ \text{fresh}(n_V, \alpha) \]

\[ (\alpha, (t, \text{send}_V(n_V))) \in R_1 \]

\[ P \in \text{Hnst} \]

\[ t \geq \max t(\alpha) \]

\[ (t', \text{recv}_P(n_V)) \ldots (n_P)) \in \alpha \]

\[ u = \text{send}_V(c) \]

\[ v = \text{recv}_V(r) \]

\[ (t_u, u) \in \alpha \]

\[ (t_v, v) \in \alpha \]

\[ r = h(\text{sh}(V,P), n_V, n_P, c) \]

\[ (\alpha, (t, \text{claim}_V(P, u, v))) \in R_5 \]

\[ \Delta t \]

\[ P \text{ is close} \]

\[ n_V \]

\[ n_P \]

\[ c \]

\[ h(k, n_V, n_P, c) \]
Hancke and Kuhn’s 2005

example

\[P = \{R_1, R_2, R_3, R_4, R_5\}\]

\[V \in Hnst \quad t \geq \maxt(\alpha)\]

\[\text{fresh}(n_V, \alpha)\]

\[(\alpha, (t, \text{send}_V(n_V))) \in R_1\]

\[P \in Hnst \quad t \geq \maxt(\alpha)\]

\[\text{fresh}(n_P, \alpha)\]

\[(\alpha, (t, \text{send}_P(n_P))) \in R_2\]

\[V \in Hnst \quad t \geq \maxt(\alpha)\]

\[(t', \text{send}_V(n_V)) \in \alpha\]

\[(t'', \text{recv}_V(n_P)) \in \alpha\]

\[\text{fresh}(c, \alpha)\]

\[(\alpha, (t, \text{send}_V(c))) \in R_3\]

\[r = h(\text{sh}(V, P), n_V, n_P, c)\]

\[(\alpha, (t, \text{claim}_V(P, u, v))) \in R_5\]
And some other stuff such as **message deduction**.

The set \( \text{infer}(A, \alpha) \) contains all messages that \( A \) can infer from \( \alpha \):

\[
\frac{m \in \text{init}(A)}{m \in \text{infer}(A, \alpha)} \quad \frac{(t, \text{recv}_A(m)) \in \alpha}{m \in \text{infer}(A, \alpha)} \quad \frac{(m_1, m_2) \in \text{infer}(A, \alpha)}{m_i \in \text{infer}(A, \alpha)}
\]

\[
\frac{m_1 \in \text{infer}(A, \alpha)}{m_2 \in \text{infer}(A, \alpha)} \quad \frac{(m_1, m_2) \in \text{infer}(A, \alpha)}{m \in \text{infer}(A, \alpha)} \quad \frac{f \in \text{Func} \setminus \{sk, -1, sh\}}{f(m) \in \text{infer}(A, \alpha)}
\]

\[
\frac{m \in \text{infer}(A, \alpha)}{k \in \text{infer}(A, \alpha)} \quad \frac{\{m\}_k \in \text{infer}(A, \alpha)}{k^{-1} \in \text{infer}(A, \alpha)} \quad \frac{m \in \text{infer}(A, \alpha)}{m \in \text{infer}(A, \alpha)}
\]
Execution model

The set of all valid traces \( Tr(\mathcal{P}) \) is defined by:

\[
\alpha \cdot (t, e) \in Tr(\mathcal{P}) \iff \alpha \in Tr(\mathcal{P}) \land \exists R \in \mathcal{P} \cup \{Int, Net\}: (\alpha, (t, e)) \in R
\]

where:

\[
\begin{align*}
I & \in Dishonest \\
t & \geq maxt(\alpha) \\
m & \in infer(I, \alpha) \\
(\alpha, (t, send_I(m))) & \in Int
\end{align*}
\]

\[
\begin{align*}
t & \geq maxt(\alpha) \\
(t', send_A(m)) & \in \alpha \\
t & \geq t' + dist(A, B) / c \\
(\alpha, (t, recv_B(m))) & \in Net
\end{align*}
\]
Secure distance-bounding

**Definition**

A protocol $\mathcal{P}$ satisfies *secure distance-bounding* if and only if:

\[ \forall \alpha \in Tr(\mathcal{P}), (t, claim_V(\mathcal{P}, u, v)) \in \alpha: \]

\[ \exists (tu, u), (tv, v) \in \alpha, \mathcal{P}' \approx \mathcal{P}: dist(V, \mathcal{P}') \leq \frac{c \cdot (tv - tu)}{2} \]

where $\approx = \{(A, A) \mid A \in Honest\} \cup Dishonest \times Dishonest$.


Jorge Toro Pozo (Univ. of Luxembourg)
1. Introduction

2. Probabilistic model based on automata

3. Symbolic model with time and location

4. Symbolic model based on causality

5. Conclusion and Future
Three timing scenarios

Correct timing
Three timing scenarios

Correct timing

Early timing

V \quad P

\Delta t

\Delta t'

chal

resp

resp

chal
Three timing scenarios

Correct timing

Early timing

Very early timing

Claim: If there is an early timing, then there is a very early timing.

Jorge Toro Pozo (Univ. of Luxembourg)
Three timing scenarios

Correct timing

Early timing

Very early timing

Claim: If there is an early timing, then there is a very early timing.
A protocol $\mathcal{P}$ satisfies secure distance-bounding if and only if:

$$\forall \sigma \in \pi(T_\mathcal{P}), \text{claim}_\mathcal{V}(P, u, v) \in \sigma :$$

$$\exists u \cdot e \cdot v \sqsubseteq \sigma : \text{actor}(e) \approx P$$

where $\pi(T) = \{e_1 \cdots e_n \mid (t_1, e_1) \cdots (t_n, e_n) \in T\}$. 
Main theorem

Theorem

A protocol $\mathcal{P}$ satisfies secure distance-bounding if and only if:

\[
\forall \sigma \in \pi(\text{Tr}(\mathcal{P})), \text{claim}_V(P, u, v) \in \sigma : \\
\exists u \cdot e \cdot v \sqsubseteq \sigma : \text{actor}(e) \approx P
\]

where $\pi(T) = \{ e_1 \cdots e_n \mid (t_1, e_1) \cdots (t_n, e_n) \in T \}$.

Proof idea
Characterise timed-traces model

For every \((t_1, e_1) \cdots (t_n, e_n) \in Tr(\mathcal{P}):\)
Proof idea
Characterise timed-traces model

For every \((t_1, e_1) \cdots (t_n, e_n) \in Tr(\mathcal{P}):\)

1. \(t_1 \leq \cdots \leq t_n\)
Proof idea
Characterise timed-traces model

For every \((t_1, e_1) \cdots (t_n, e_n) \in Tr(P)\):

1. \(t_1 \leq \cdots \leq t_n\)

2. \(t_n = recv_A(m)\) implies \(i < n\) exists such that \(e_i = send_B(m)\) and \(t_n - t_i \geq dist(A, B) / c\)
Proof idea
Characterise timed-traces model

For every \((t_1, e_1) \cdots (t_n, e_n) \in Tr(P)\):

1. \(t_1 \leq \cdots \leq t_n\)
2. \(t_n = recv_A(m)\) implies \(i < n\) exists such that \(e_i = send_B(m)\) and \(t_n - t_i \geq dist(A, B)/c\)
3. if \((t'_1, e_1) \cdots (t'_n, e_n)\) satisfies (1) and (2) then \((t'_1, e_1) \cdots (t'_n, e_n) \in Tr(P)\)
Proof idea
Characterise timed-traces model

For every $e_1 \cdots e_n \in \pi(\text{Tr}(\mathcal{P}))$:
Proof idea
Characterise timed-traces model

For every $e_1 \cdots e_n \in \pi(\text{Tr}(\mathcal{P}))$:

1. $e_1 \cdots e_{n-1} \in \pi(\text{Tr}(\mathcal{P}))$
Proof idea
Characterise timed-traces model

For every $e_1 \cdots e_n \in \pi(Tr(P))$:

1. $e_1 \cdots e_{n-1} \in \pi(Tr(P))$

2. $e_n \not\in \text{Recv}$ and $\text{actor}(e_{n-1}) \neq \text{actor}(e_n)$ then
   $e_1 \cdots e_{n-2} \cdot e_n \in \pi(Tr(P))$
Proof idea
Characterise timed-traces model

For every \( e_1 \cdots e_n \in \pi(Tr(P)) \):

1. \( e_1 \cdots e_{n-1} \in \pi(Tr(P)) \)

2. \( e_n \notin \text{Recv} \) and \( \text{actor}(e_{n-1}) \neq \text{actor}(e_n) \) then \( e_1 \cdots e_{n-2} \cdot e_n \in \pi(Tr(P)) \)

3. \( e_n = \text{send}_A(m) \) implies \( e_1 \cdots e_n \cdot \text{recv}_B(m) \in \pi(Tr(P)) \)
Proof idea
Characterise timed-traces model

For every $e_1 \cdots e_n \in \pi(Tr(P))$:

1. $e_1 \cdots e_{n-1} \in \pi(Tr(P))$

2. $e_n \not\in \text{Recv}$ and $\text{actor}(e_{n-1}) \neq \text{actor}(e_n)$ then $e_1 \cdots e_{n-2} \cdot e_n \in \pi(Tr(P))$

3. $e_n = \text{send}_A(m)$ implies $e_1 \cdots e_n \cdot \text{recv}_B(m) \in \pi(Tr(P))$

4. $\forall A, B \in \text{Honest}^2 \cup \text{Dishonest}^2$ it holds that $(e_1 \cdots e_n)[A \leftrightarrow B] \in \pi(Tr(P))$
The Tamarin dbsec lemma

lemma dbsec:
"
All P V m n #t. (VerifierComplete(P, V, m, n)@t ) =>
(Ex #tc.
  Corrupt(V)@tc
) |(
  Ex #t1 #t2 #t3.
  StartFastPhase(V, m)@t1 &
  Action(P)@t2 &
  EndFastPhase(V, m)@t3 &
  (#t1 < #t2) &
  (#t2 < #t3) &
  ((#t3 < #t) | (#t3 = #t))
)
 |(
  Ex CAgent #t4 #t5 #t6 #t7.
  StartFastPhase(V, m)@t5 &
  EndFastPhase(V, m)@t7 &
  Corrupted(P, V)@t4 &
  CAction(CAgent)@t6 &
  (#t5 < #t6) &
  (#t6 < #t7) &
  ((#t7 < #t) | (#t7 = #t))
)
"

## Verification in Tamarin

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Satisfies dbsec?</th>
<th>Attack found</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC-Signature</td>
<td>No</td>
<td>DH</td>
</tr>
<tr>
<td>BC-FiatShamir</td>
<td>No</td>
<td>DH, DF</td>
</tr>
<tr>
<td>BC-Schnorr</td>
<td>No</td>
<td>DH, DF</td>
</tr>
<tr>
<td>CRCS</td>
<td>No</td>
<td>DH</td>
</tr>
<tr>
<td>Meadows et al.</td>
<td>No</td>
<td>DH</td>
</tr>
<tr>
<td>Tree-based</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Poulidor</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Hancke and Kuhn</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Uniform</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Kim and Avoine</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Munilla et al.</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Reid et al.</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>Swiss-Knife</td>
<td>Yes</td>
<td>-</td>
</tr>
<tr>
<td>TREAD-PK</td>
<td>No</td>
<td>MF, DH</td>
</tr>
<tr>
<td>TREAD-SH</td>
<td>No</td>
<td>DH</td>
</tr>
<tr>
<td>PaySafe</td>
<td>No</td>
<td>DF, DH</td>
</tr>
</tbody>
</table>
What we achieved

- Proved that secure distance-bounding can be formulated through causality.
What we achieved

- Proved that secure distance-bounding can be formulated through causality.

- Provided a fully-automatic verification framework for DB protocols. (simply specify the protocol and click on “verify dbsec lemma”).

Identification of unreported vulnerabilities in two published protocols: PaySafe (FC’15) and TREAD (AsiaCCS’17).
What we achieved

- Proved that secure distance-bounding can be formulated through causality.

- Provided a fully-automatic verification framework for DB protocols. (simply specify the protocol and click on “verify dbsec lemma”).

- Provided computer-verifiable (in)security proofs for a number of state-of-the-art protocols.
What we achieved

- Proved that secure distance-bounding can be formulated through causality.

- Provided a fully-automatic verification framework for DB protocols. (simply specify the protocol and click on “verify dbsec lemma”).

- Provided computer-verifiable (in)security proofs for a number of state-of-the-art protocols.

- Identified unreported vulnerabilities in two published protocols: PaySafe (FC’15) and TREAD (AsiaCCS’17).
Outline

1. Introduction
2. Probabilistic model based on automata
3. Symbolic model with time and location
4. Symbolic model based on causality
5. Conclusion and Future
Probabilistic vs. Symbolic

- Where probabilistic models win:
  - More precise results - there’s an attach that succeeds with prob. $p$
  - Arithmetic properties can be fairly-well modeled

- Where symbolic models win:
  - No need to consider each attack individually
  - Automated verification - Tamarin, ProVerif, Scyther, Isabelle
  - Computer-verifiable proofs of (in)security
Future

- Terrorist fraud?
  Requires fancy techniques for corruption modeling.
Future

- **Terrorist fraud?**
  Requires fancy techniques for corruption modeling.

- **Automatic probabilistic analysis?**
  Seems hard.
Thank you

jorge.toro@uni.lu
http://satoss.uni.lu/jorge