LTL Model Checking for Security Protocols

Roberto Carbone\(^1\)
(joint work with A. Armando\(^1\) and L. Compagna\(^2\))

\(^1\)Artificial Intelligence Laboratory, DIST, University of Genova
\(^2\)SAP Research, Sophia-Antipolis, France

20th IEEE Computer Security Foundations Symposium
Venice - July 6-8, 2007
Outline

1. Introduction
2. LTL Model Checking for Security Protocols
3. A Motivating Example: the ASW Protocol
   - The Protocol
   - The Assumptions on the Channels and on the Honest Principals
   - Security Properties
4. Modelling, Specification and Analysis using Set-rewriting and LTL
   - Specifying the Protocol Scenario and the Intruder
   - Specifying the Assumptions on the Channels and on the Honest Principals
   - Specifying Security Properties
5. Conclusions
Motivations

Most security protocol analysers assume that:

1. **channels** controlled by DY intruder
2. **honest agents** are only asked to react to received messages by sending some other messages.
3. **security properties** are expressed as reachability properties

In many cases these assumptions can be circumvented by **transforming the problem** (i.e. model+property) at hand.

In some cases these transformations can be automated.

In other, more complex cases this must be done manually, but this is in general **difficult** and hence **time consuming** and **error prone**.
Most security protocol analysers assume that:

1. **channels** controlled by DY intruder
2. **honest agents** are **only** asked to react to received messages by sending some other messages.
3. **security properties** are expressed as reachability properties

In many cases these assumptions can be circumvented by **transforming the problem** (i.e. model+property) at hand.

In some cases these transformations can be automated.

In other, more complex cases this must be done manually, but this is in general **difficult** and hence **time consuming** and **error prone**.
Motivations: a concrete example

The optimistic fair exchange protocol proposed by Asokan, Shoup, and Waidner (ASW) violates all three assumptions:

- **Channels** are assumed to be confidential and/or resilient.
- **Agents** are assumed to make progress during the execution of the protocol ($O$ and $R$ must timeout and $T$ must be always available).
- **Security property** (fair exchange) cannot be directly expressed as a reachability property.
general framework for automatic security protocol analysis that, by using

- set-rewriting for specifying the model and
- LTL for specifying security properties and constraints,

allows us to relieve the above assumptions

extension of SATMC, a SAT-based model checker for security protocols, to support the analysis of LTL formulae and

effectiveness of the approach assessed through analysis of the ASW protocol: new attack found and key security property refined.
Outline

1. Introduction

2. LTL Model Checking for Security Protocols

3. A Motivating Example: the ASW Protocol
   - The Protocol
   - The Assumptions on the Channels and on the Honest Principals
   - Security Properties

4. Modelling, Specification and Analysis using Set-rewriting and LTL
   - Specifying the Protocol Scenario and the Intruder
   - Specifying the Assumptions on the Channels and on the Honest Principals
   - Specifying Security Properties

5. Conclusions
Security protocol analysis boils down to building and solving model checking problems of the form:

\[ M \models (C_I \land C_H) \supset G \]

where

- **M**: transition system modelling a superset of the behaviours of the honest agents and of the intruder.
- **C_I**: the allowed behaviours of the intruder.
- **C_H**: the allowed behaviours of the honest principals.
- **G**: encoding the security property.
Security protocol analysis boils down to building and solving model checking problems of the form:

\[ M \models (C_I \land C_H) \supset G \]

where

- **M**: transition system modelling a superset of the behaviours of the honest agents and of the intruder.
- **C_I**: the allowed behaviours of the intruder.
- **C_H**: the allowed behaviours of the honest principals.
- **G**: encoding the security property.
Security protocol analysis boils down to building and solving model checking problems of the form:

\[ M \models (C_I \land C_H) \supset G \]

where

- **M**: transition system modelling a superset of the behaviours of the honest agents and of the intruder.
- **C_I**: the allowed behaviours of the intruder.
- **C_H**: the allowed behaviours of the honest principals.
- **G**: encoding the security property.
Security protocol analysis boils down to building and solving model checking problems of the form:

\[ M \models (C_I \land C_H) \supset G \]

where

- \( M \): transition system modelling a superset of the behaviours of the honest agents and of the intruder.
- \( C_I \): the allowed behaviours of the intruder.
- \( C_H \): the allowed behaviours of the honest principals.
- \( G \): encoding the security property.
SATMC: SAT-based Model Checking of Security Protocols

- SATMC reduces the problem of determining whether a protocol violates a security property in $k$ steps to SAT.

- By leveraging on state-of-the-art SAT solvers, SATMC can compete with—and in some cases outperform—other state-of-the-art protocol analysers.

- SATMC is a back-end of the AVISPA Tool.

- We have extended SATMC to support model checking of LTL formulae.
Outline

1 Introduction

2 LTL Model Checking for Security Protocols

3 A Motivating Example: the ASW Protocol
   - The Protocol
   - The Assumptions on the Channels and on the Honest Principals
   - Security Properties

4 Modelling, Specification and Analysis using Set-rewriting and LTL
   - Specifying the Protocol Scenario and the Intruder
   - Specifying the Assumptions on the Channels and on the Honest Principals
   - Specifying Security Properties

5 Conclusions
The ASW Protocol

The ASW protocol consists of three subprotocols.

The exchange subprotocol: played by $O$ and $R$ and—if successful—it allows to achieve a mutual commitment on a previously agreed contractual text (standard contract).

The abort subprotocol: $O$ can request $T$ to abort the previously initiated contract signing procedure.

The resolve subprotocol: $O$ and $R$ can request $T$ to force the resolution of the contract, possibly obtaining a replacement contract.
The ASW protocol consists of three subprotocols.

The exchange subprotocol: played by $O$ and $R$ and—if successful—it allows to achieve a mutual commitment on a previously agreed contractual text (standard contract).

The abort subprotocol: $O$ can request $T$ to abort the previously initiated contract signing procedure.

The resolve subprotocol: $O$ and $R$ can request $T$ to force the resolution of the contract, possibly obtaining a replacement contract.
The ASW Protocol

The ASW protocol consists of three subprotocols.

The exchange subprotocol: played by $O$ and $R$ and—if successful—it allows to achieve a mutual commitment on a previously agreed contractual text (standard contract).

The abort subprotocol: $O$ can request $T$ to abort the previously initiated contract signing procedure.

The resolve subprotocol: $O$ and $R$ can request $T$ to force the resolution of the contract, possibly obtaining a replacement contract.
The ASW Protocol

The ASW protocol consists of three subprotocols.

The exchange subprotocol: played by \( O \) and \( R \) and—if successful—it allows to achieve a mutual commitment on a previously agreed contractual text (standard contract).

The abort subprotocol: \( O \) can request \( T \) to abort the previously initiated contract signing procedure.

The resolve subprotocol: \( O \) and \( R \) can request \( T \) to force the resolution of the contract, possibly obtaining a replacement contract.
The Exchange Subprotocol

**First round:** $O$ and $R$ express their public commitments to the contract.

**Second round:** $O$ and $R$ exchange their secret commitments, required for the standard contract (i.e. $\{me_1, NO, me_2, NR\}$).

\[E1. \quad O \rightarrow R : \quad me_1 = \text{Sig}_O(V_O, V_R, T, \text{Text}, h(NO))\]

\[E2. \quad R \rightarrow O : \quad me_2 = \text{Sig}_R(me_1, h(NR))\]

\[E3. \quad O \rightarrow R : \quad NO\]

\[E4. \quad R \rightarrow O : \quad NR\]
The Exchange Subprotocol

First round: $O$ and $R$ express their public commitments to the contract.

Second round: $O$ and $R$ exchange their secret commitments, required for the standard contract (i.e. $\{me_1, NO, me_2, NR\}$).

\[ E1. \quad O \rightarrow R \quad : \quad me_1 = \text{Sig}_O(V_O, V_R, T, \text{Text}, h(NO)) \]

\[ E2. \quad R \rightarrow O \quad : \quad me_2 = \text{Sig}_R(me_1, h(NR)) \]

\[ E3. \quad O \rightarrow R \quad : \quad NO \]

\[ E4. \quad R \rightarrow O \quad : \quad NR \]
A1. $O \rightarrow T : ma_1 = \text{Sig}_O(aborted, me_1)$

A2. $T \rightarrow O :$

if $\text{resolved}(me_1, me_2) \in DB$
then $\text{Sig}_T(me_1, me_2)$
// Replacement contract
else $DB := DB \cup \{\text{aborted}(me_1)\}$;
$\text{Sig}_T(aborted, ma_1)$
// Abort token
The Abort Subprotocol

A1. $O \rightarrow T : \quad ma_1 = \text{Sig}_O(\text{aborted}, me_1)$

A2. $T \rightarrow O : \quad$ if $\text{resolved}(me_1, me_2) \in DB$
then $\text{Sig}_T(me_1, me_2)$
  // Replacement contract
else $DB := DB \cup \{\text{aborted}(me_1)\};$
  $\text{Sig}_T(\text{aborted}, ma_1)$
  // Abort token
A1. \( O \rightarrow T \) : \( ma_1 = \text{Sig}_O(aborted, me_1) \)

A2. \( T \rightarrow O \) : if \( \text{resolved}(me_1, me_2) \in DB \)
then \( \text{Sig}_T(me_1, me_2) \)
// Replacement contract
else \( DB := DB \cup \{\text{aborted}(me_1)\} \);
\( \text{Sig}_T(aborted, ma_1) \)
// Abort token
The Resolve Subprotocol (1)

\[ R_1. \quad R \rightarrow T \quad : \quad mr_1 = \langle me_1, me_2 \rangle \]

\[ R_2. \quad T \rightarrow R \quad : \quad \begin{cases} 
\text{if } \text{aborted}(me_1) \in DB \\
\text{then } \text{Sig}_T(aborted, ma_1) \\
\quad \text{// Abort token} \\
\text{else } DB := DB \cup \{\text{resolved}(me_1, me_2)\}; \\
\quad \text{Sig}_T(me_1, me_2) \\
\quad \text{// Replacement contract} 
\end{cases} \]
The Resolve Subprotocol (1)

R1. \( R \rightarrow T \) : \( mr_1 = \langle me_1, me_2 \rangle \)

R2. \( T \rightarrow R \) : if aborted(\( me_1 \)) \( \in \) \( DB \)

then \( \text{Sig}_T(aborted, ma_1) \)

// Abort token

else \( DB := DB \cup \{\text{resolved}(me_1, me_2)\} \);

\( \text{Sig}_T(me_1, me_2) \)

// Replacement contract
The Resolve Subprotocol (1)

$R_1$. $R \to T : mr_1 = \langle me_1, me_2 \rangle$

$R_2$. $T \to R$ : if aborted$(me_1) \in DB$

then $\text{Sig}_T(aborted, ma_1)$

// Abort token

else $DB := DB \cup \{\text{resolved}(me_1, me_2)\}$

$\text{Sig}_T(me_1, me_2)$

// Replacement contract
The Resolve Subprotocol (2)

R1. \( O \to T \) : \( mr_1 = \langle me_1, me_2 \rangle \)

R2. \( T \to O \) : if \( \text{aborted}(me_1) \in DB \) then \( \text{Sig}_T(\text{aborted}, ma_1) \)

// Abort token
else \( DB := DB \cup \{\text{resolved}(me_1, me_2)\} \);
\( \text{Sig}_T(me_1, me_2) \)

// Replacement contract
Confidential: eavesdroppers do not have access to the information.

Resilient: any message will be eventually delivered to the intended recipient.
**Confidential**: eavesdroppers do not have access to the information.

**Resilient**: any message will be eventually delivered to the intended recipient.
ASW: Assumptions on the Honest Principals

- **Timeout (O and R):** During the exchange sub-protocol, O and R will not indefinitely wait for a reply from the corresponding party and will eventually start either the abort or resolve subprotocol.

- **Availability (T):** T must be always available, i.e. he must eventually process all the messages received by replying to requests by O and R.
Fair exchange can be expressed as the conjunction of the following two properties (slightly adapted from [SM02]):

(A) A corrupt principal cannot obtain a valid contract without allowing the remaining principal to also obtain a valid contract.

(B) Once an honest principal obtains an abort token, it is impossible for any other principal to obtain a valid contract.

Outline

1. Introduction
2. LTL Model Checking for Security Protocols
3. A Motivating Example: the ASW Protocol
   - The Protocol
   - The Assumptions on the Channels and on the Honest Principals
   - Security Properties
4. Modelling, Specification and Analysis using Set-rewriting and LTL
   - Specifying the Protocol Scenario and the Intruder
   - Specifying the Assumptions on the Channels and on the Honest Principals
   - Specifying Security Properties
5. Conclusions
The Model

\[ M \models (C_I \land C_H) \supset G \]

Transition system associated with the concurrent execution of a number of sessions of the protocol.

- States: sets of facts, i.e. ground atomic formulae
- Transitions: rewrite rules that define mappings between sets of facts.

If \( S \) is a set of facts, then we interpret the facts in \( S \) as the propositions holding in the state represented by \( S \), all other facts being false in that state (closed-world assumption).
### The Model: Facts

<table>
<thead>
<tr>
<th>Fact</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{state}_{\text{Role}}(j, a, es, s)</td>
<td>Principal $a$, playing role $\text{Role}$, is ready to execute step $j$ in session $s$ of the protocol, and $es$ is a list of expressions representing the internal state of $a$ and thus affecting her future behaviour.</td>
</tr>
<tr>
<td>$\text{ak}(a, m)$</td>
<td>Principal $a$ knows message $m$.</td>
</tr>
<tr>
<td>$\text{sent}(rs, b, a, m, c)$</td>
<td>Principal $rs$ has sent message $m$ on channel $c$ to principal $a$ pretending to be principal $b$.</td>
</tr>
<tr>
<td>$\text{rcvd}(a, b, m, c)$</td>
<td>Message $m$ (supposedly sent by principal $b$) has been received on channel $c$ by principal $a$, but $a$ has not processed it yet.</td>
</tr>
</tbody>
</table>

Note: $\text{ik}(m)$ abbreviates $\text{ak}(i, m)$.  

---

Roberto Carbone (AI-Lab)
The Model: Rules for the Honest Agents

Delivery of messages:

\[
\text{sent}(RS, B, A, M, C) \quad \xrightarrow{\text{receive}(A,B,RS,M,C)} \quad \text{rcvd}(A, B, M, C) \]

Processing a previously received message:

\[
\text{rcvd}(A, B, M, C) \quad \xrightarrow{\text{send}_{j}(A,B,B1,...,S)} \quad \text{sent}(A, A, B1, M1, C1) \]

\[
\text{state}_{Role}(j, A, es, S) \quad \xrightarrow{\text{send}_{j}(A,B,B1,...,S)} \quad \text{state}_{Role}(l, A, es', S) \]
Delivery of messages:

\[
\text{sent}(RS, B, A, M, C) \quad \text{receive}(A, B, RS, M, C) \quad \text{rcvd}(A, B, M, C) \quad . \text{ak}(A, M)
\]

Processing a previously received message:

\[
\text{rcvd}(A, B, M, C) \quad \text{state}_{Role}(j, A, es, S) \quad \text{send}_{j}(A, B, B1, ..., S) \quad \text{sent}(A, A, B1, M1, C1) \quad \text{state}_{Role}(l, A, es', S)
\]
The Model: Rules for the Intruder

\[\text{sent}(A, A, B, M, C) \xrightarrow{\text{intercept}(A, B, M, C)} \text{rcvd}(i, A, M, C) \cdot \text{ik}(M)\]

\[\text{sent}(A, A, B, M, C) \xrightarrow{\text{overhear}(A, B, M, C)} \text{sent}(A, A, B, M, C) \cdot \text{rcvd}(i, A, M, C) \cdot \text{ik}(M)\]

\[\text{ik}(M) \cdot \text{ik}(A) \cdot \text{ik}(B) \xrightarrow{\text{fake}(A, B, M, C)} \text{sent}(i, A, B, M, C) \cdot \text{ik}(M) \cdot \text{ik}(A) \cdot \text{ik}(B)\]
\[\text{ak}(A, M) \cdot \text{ak}(A, K) \xrightarrow{\text{encrypt}(A,K,M)} \text{ak}(A, M) \cdot \text{ak}(A, K) \cdot \text{ak}(A, \{M\}_K)\]

\[\text{ak}(A, \{M\}_K) \cdot \text{ak}(A, K^{-1}) \xrightarrow{\text{decrypt}_\text{puk}(A,K,M)} \text{ak}(A, \{M\}_K) \cdot \text{ak}(A, K^{-1}) \cdot \text{ak}(A, M)\]

\[\text{ak}(A, \{M\}_{K^{-1}}) \cdot \text{ak}(A, K) \xrightarrow{\text{decrypt}_\text{prk}(A,K,M)} \text{ak}(A, \{M\}_{K^{-1}}) \cdot \text{ak}(A, K) \cdot \text{ak}(A, M)\]

\[\text{ak}(A, M_1) \cdot \text{ak}(A, M_2) \xrightarrow{\text{pairing}(A,M_1,M_2)} \text{ak}(A, M_1) \cdot \text{ak}(A, M_2) \cdot \text{ak}(A, \langle M_1, M_2 \rangle)\]

\[\text{ak}(A, \langle M_1, M_2 \rangle) \xrightarrow{\text{decompose}(A,M_1,M_2)} \text{ak}(A, \langle M_1, M_2 \rangle) \cdot \text{ak}(A, M_1) \cdot \text{ak}(A, M_2)\]
Constraining the Behaviour of the Intruder

\[ M \models (C_I \land C_H) \supset G \]

- The **confidentiality** of channel \( c \) to a set of principals \( PS \) is formalised by:

\[ G \forall (rcvd(B, A, M, c) \supset \bigvee_{p \in PS} B = p) \]

- The **resilience** of channel \( c \) is formalised by:

\[ G \forall (sent(A', A, B, M, c) \supset F rcvd(B, A, M, c)) \]

...
Constraining the Behaviour of Honest Principals

\[ M \models (C_I \land C_H) \supset G \]

- \( O \) and \( R \) should not indefinitely wait for an answer:

\[ G \forall (\text{state}_{Role}(j, \ldots) \supset F \neg \text{state}_{Role}(j, \ldots)) \]

where \( j \) identifies the states from which a time-out can occur.

- Messages received will be eventually processed by \( T \)

\[ G \forall (\text{rcvd}(t, P, M, C) \supset F \neg \text{rcvd}(t, P, M, C)) \]
Fair Exchange

(A) A corrupt principal cannot obtain a valid contract without allowing the remaining principal to also obtain a valid contract.

(B) Once an honest principal obtains an abort token, it is impossible for any other principal to obtain a valid contract.
M |= (C_I \land C_H) \supset G_A

Fair Exchange

(A) A corrupt principal cannot obtain a valid contract without allowing the remaining principal to also obtain a valid contract.

(B) Once an honest principal obtains an abort token, it is impossible for any other principal to obtain a valid contract.
Specifying Security Properties: Fair Exchange

\[ M \models (C_I \land C_H) \supset G_B \]

**Fair Exchange**

(A) A corrupt principal cannot obtain a valid contract without allowing the remaining principal to also obtain a valid contract.

(B) Once an honest principal obtains an abort token, it is impossible for any other principal to obtain a valid contract.
Specifying and Analysing Fair Exchange (A)
Fair Exchange (A): Specification

\[ M \models (C_i \land C_H) \supset G_A \]

(A) A corrupt principal cannot obtain a valid contract without allowing the remaining principal to also obtain a valid contract.

\( G_A \) is the conjunction, for each session considered, of all formulae of the form:

\[ G \forall (hasvc(o, o, r, txt, N_O, N_R, t) \supset F hasvc(r, o, r, txt, N_O, N_R, t)) \]

If \( o \) has a valid contract, binding \( r \) to \( txt \) using \( N_O \) and \( N_R \) as secret commitments and \( t \) as \( T \), then eventually \( r \) will have a corresponding valid contract.
Fair Exchange (A): Specification

\[ M \models (C_i \land C_H) \supset G_A \]

(A) A corrupt principal cannot obtain a valid contract without allowing the remaining principal to also obtain a valid contract.

\[ G_A \] is the conjunction, for each session considered, of all formulae of the form:

\[ G \forall (\mathit{hasvc}(o, o, r, \mathit{txt}, N_O, N_R, t) \supset F \mathit{hasvc}(r, o, r, \mathit{txt}, N_O, N_R, t)) \]

If \( o \) has a valid contract, binding \( r \) to \( \mathit{txt} \) using \( N_O \) and \( N_R \) as secret commitments and \( t \) as \( T \), then eventually \( r \) will have a corresponding valid contract.
Fair Exchange (A): Specification

\[ M \models (C_i \land C_H) \supset G_A \]

(A) A corrupt principal cannot obtain a valid contract without allowing the remaining principal to also obtain a valid contract.

\( G_A \) is the conjunction, for each session considered, of all formulae of the form:

\[ G \forall (\text{hasvc}(o, o, r, txt, N_O, N_R, t) \supset F\text{hasvc}(r, o, r, txt, N_O, N_R, t)) \]

If \( o \) has a valid contract, binding \( r \) to \( txt \) using \( N_O \) and \( N_R \) as secret commitments and \( t \) as \( T \), then eventually \( r \) will have a corresponding valid contract.
Fair Exchange (A): Specification

\[ \text{hasvc}(P, O, R, \text{Txt}, N_O, N_R, T) \equiv \]

\[ \begin{align*}
\text{// } P \text{ has a standard contract, i.e. he knows } me_1, me_2, N_O, \text{ and } N_R \\
\text{ak}(P, me_1(O, R, \text{Txt}, N_O, T)) \land \\
\text{ak}(P, me_2(O, R, \text{Txt}, N_O, N_R, T)) \land \\
\text{ak}(P, N_O) \land \\
\text{ak}(P, N_R) \\
\lor \\
\text{// } P \text{ has a replacement contract, i.e. he knows } \text{Sig}_T(me_1, me_2) \\
\text{ak}(P, \text{Sig}_T(me_1(O, R, \text{Txt}, N_O, T), me_2(O, R, \text{Txt}, N_O, N_R, T)))
\end{align*} \]
hasvc\((P, O, R, \text{Txt}, N_O, N_R, T)\) :=

// P has a standard contract, i.e. he knows \(me_1, me_2, N_O, \text{and } N_R\)
\[\text{ak}(P, me_1(O, R, \text{Txt}, N_O, T)) \land \text{ak}(P, me_2(O, R, \text{Txt}, N_O, N_R, T)) \land \text{ak}(P, N_O) \land \text{ak}(P, N_R)\]

\lor

// P has a replacement contract, i.e. he knows \(\text{Sig}_T(me_1, me_2)\)
\[\text{ak}(P, \text{Sig}_T(me_1(O, R, \text{Txt}, N_O, T), me_2(O, R, \text{Txt}, N_O, N_R, T)))\]
\[\text{hasvc}(P, O, R, \text{Txt}, N_O, N_R, T) :=\]

\[
\begin{align*}
\text{// } P \text{ has a standard contract, i.e. he knows } me_1, me_2, N_O, \text{ and } N_R \\
[ & \text{ak}(P, me_1(O, R, \text{Txt}, N_O, T)) \land \\
& \text{ak}(P, me_2(O, R, \text{Txt}, N_O, N_R, T)) \land \\
& \text{ak}(P, N_O) \land \\
& \text{ak}(P, N_R) ] \\
\lor \\
\text{// } P \text{ has a replacement contract, i.e. he knows } \text{Sig}_T(me_1, me_2) \\
& \text{ak}(P, \text{Sig}_T(me_1(O, R, \text{Txt}, N_O, T), me_2(O, R, \text{Txt}, N_O, N_R, T)))
\end{align*}
\]
SATMC finds the following attack:

\begin{align*}
E1. & \quad O \rightarrow I : \quad me_1 \\
E2. & \quad I \rightarrow O : \quad me_2 \\
& \quad I \text{ computes new random } N'_I \text{ and then } me'_2 \\
E3. & \quad O \rightarrow I : \quad NO \\
E4. & \quad I \rightarrow O : \quad NI
\end{align*}

At the end of the protocol, the intruder owns a standard contract 
\{me_1, me'_2, NO, N'_I\} while O owns only the standard contract 
\{me_1, me_2, NO, NI\}.

This attack is similar in spirit to the attack in [SM02], but it is simpler as \(T\) is not involved.
SATMC finds the following attack:

\[
\begin{align*}
E1. & \quad O \rightarrow I : \quad me_1 \\
E2. & \quad I \rightarrow O : \quad me_2 \\
\quad & \quad I \text{ computes new random } N'_I \text{ and then } me'_2 \\
E3. & \quad O \rightarrow I : \quad N_O \\
E4. & \quad I \rightarrow O : \quad N_I
\end{align*}
\]

At the end of the protocol, the intruder owns a standard contract \(\{me_1, me'_2, N_O, N'_I\}\) while \(O\) owns only the standard contract \(\{me_1, me_2, N_O, N_I\}\).

This attack is similar in spirit to the attack in [SM02], but it is simpler as \(T\) is not involved.
Fair Exchange (A): Analysis of the Patched Version

[SM02] propose to repair ASW by replacing steps $E3$ and $E4$

$E3. \ O \rightarrow \ R \ : \ N_O$
$E4. \ R \rightarrow \ O \ : \ N_R$

SATMC confirms that the improved version of the protocol does not suffer from the previous attack, but it detects a new (i.e. previously unknown) attack:

$E1. \ \ O \rightarrow \ I \ : \ me_1$
$E2. \ I \rightarrow \ O \ : \ me_2$
$I$ computes new random $N'_I$ and then $me'_2$
$E3'. \ O \rightarrow \ I \ : \ \text{Sig}_O(N_O, h(N_I))$
$E4'. \ I \rightarrow \ O \ : \ \text{Sig}_I(N_I, h(N_O))$
$R1. \ I \rightarrow \ T \ : \ mr_1 = \langle me_1, me'_2 \rangle$
$R2. \ T \rightarrow \ I \ : \ mr_2 = \text{Sig}_T(me_1, me'_2)$

The intruder, here playing the responder, obtains a replacement contract relative to the secret commitments $N_O$ and $N'_I$, while $O$ obtains only a valid contract relative to the secret commitments $N_O$ and $N_I$. 
[SM02] propose to repair ASW by replacing steps $E3$ and $E4$ with:

\[
\begin{align*}
E3' & : O \rightarrow R : \text{Sig}_O(N_O, h(N_R)) \\
E4' & : R \rightarrow O : \text{Sig}_R(N_R, h(N_O))
\end{align*}
\]

SATMC confirms that the improved version of the protocol does not suffer from the previous attack, but it detects a new (i.e. previously unknown) attack:

\[
\begin{align*}
E1 & : O \rightarrow I : \text{me}_1 \\
E2 & : I \rightarrow O : \text{me}_2 \\
 & \quad \text{I computes new random } N'_I \text{ and then me}'_2 \\
E3' & : O \rightarrow I : \text{Sig}_O(N_O, h(N_I)) \\
E4' & : I \rightarrow O : \text{Sig}_I(N_I, h(N_O)) \\
R1 & : I \rightarrow T : mr_1 = \langle \text{me}_1, \text{me}'_2 \rangle \\
R2 & : T \rightarrow I : mr_2 = \text{Sig}_T(\text{me}_1, \text{me}'_2)
\end{align*}
\]

The intruder, here playing the responder, obtains a replacement contract relative to the secret commitments $N_O$ and $N'_I$, while $O$ obtains only a valid contract relative to the secret commitments $N_O$ and $N_I$. 
[SM02] propose to repair ASW by replacing steps $E3$ and $E4$ with:

\[
E3' \quad O \rightarrow R \; : \; \text{Sig}_O(N_O, h(N_R)) \\
E4' \quad R \rightarrow O \; : \; \text{Sig}_R(N_R, h(N_O))
\]

SATMC confirms that the improved version of the protocol does not suffer from the previous attack, but it detects a new (i.e. previously unknown) attack:

\[
E1 \quad O \rightarrow I \; : \; me_1 \\
E2 \quad I \rightarrow O \; : \; me_2 \\
\text{I computes new random } N'_I \text{ and then } me'_2 \\
E3' \quad O \rightarrow I \; : \; \text{Sig}_O(N_O, h(N_I)) \\
E4' \quad I \rightarrow O \; : \; \text{Sig}_I(N_I, h(N_O)) \\
R1 \quad I \rightarrow T \; : \; mr_1 = \langle me_1, me'_2 \rangle \\
R2 \quad T \rightarrow I \; : \; mr_2 = \text{Sig}_T(me_1, me'_2)
\]

The intruder, here playing the responder, obtains a replacement contract relative to the secret commitments $N_O$ and $N'_I$, while $O$ obtains only a valid contract relative to the secret commitments $N_O$ and $N_I$. 
[SM02] propose to repair ASW by replacing steps $E3$ and $E4$ with:

$$
E3'. O \rightarrow R : \text{Sig}_O(N_O, h(N_R)) \\
E4'. R \rightarrow O : \text{Sig}_R(N_R, h(N_O))
$$

SATMC confirms that the improved version of the protocol does not suffer from the previous attack, but it detects a new (i.e. previously unknown) attack:

$$
E1. O \rightarrow I : \text{me}_1 \\
E2. I \rightarrow O : \text{me}_2 \\
I \text{ computes new random } N'_I \text{ and then } me'_2 \\
E3'. O \rightarrow I : \text{Sig}_O(N_O, h(N_I)) \\
E4'. I \rightarrow O : \text{Sig}_I(N_I, h(N_O)) \\
R1. I \rightarrow T : mr_1 = \langle \text{me}_1, me'_2 \rangle \\
R2. T \rightarrow I : mr_2 = \text{Sig}_T(\text{me}_1, me'_2)
$$

The intruder, here playing the responder, obtains a replacement contract relative to the secret commitments $N_O$ and $N'_I$, while $O$ obtains only a valid contract relative to the secret commitments $N_O$ and $N_I$. 
[SM02] propose to repair ASW by replacing steps $E3$ and $E4$ with:

$$
E3'. \ O \rightarrow \ R : \ Sig_O(N_O, h(N_R)) \\
E4'. \ R \rightarrow \ O : \ Sig_R(N_R, h(N_O))
$$

SATMC confirms that the improved version of the protocol does not suffer from the previous attack, but it detects a new (i.e. previously unknown) attack:

$$
E1. \ O \rightarrow I : \ me_1 \\
E2. \ I \rightarrow O : \ me_2 \\
\text{I computes new random } N'_I \text{ and then } me'_2 \\
E3'. \ O \rightarrow I : \ Sig_O(N_O, h(N_I)) \\
E4'. \ I \rightarrow O : \ Sig_I(N_I, h(N_O)) \\
R1. \ I \rightarrow T : \ mr_1 = \langle me_1, me'_2 \rangle \\
R2. \ T \rightarrow I : \ mr_2 = Sig_T(me_1, me'_2)
$$

The intruder, here playing the responder, obtains a replacement contract relative to the secret commitments $N_O$ and $N'_I$, while $O$ obtains only a valid contract relative to the secret commitments $N_O$ and $N_I$. 
Also this new attack does not have serious consequences: $O$ and $R$ do not have the same contract, but they have a contract for the same text.

Yet the ability to detect this attack, which has eluded previous formal analyses of the protocol, gives evidence of the effectiveness of the approach.

We have come to the conclusion that the problem is not in the protocol but in the property, which requires the contract to be relative to the same secret commitment $N_R$.

We have therefore weakened the property accordingly and successfully checked it with SATMC.
Specifying and Analysing
Fair Exchange (B)
Fair Exchange (B): Specification

\[ M \models (C_i \land C_H) \supset G_B \]

(B) Once an honest principal obtains an abort token, it is impossible for any other principal to obtain a valid contract.

We rephrase the property and express it in LTL:

(B') If an honest principal obtains an abort token, then any other principal has already a corresponding valid contract or will not be able to obtain one in the future.
SATMC finds the following attack:

<table>
<thead>
<tr>
<th>Step</th>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1.</td>
<td>$I \rightarrow R$</td>
<td>$me_1$</td>
</tr>
<tr>
<td>E2.</td>
<td>$R \rightarrow I$</td>
<td>$me_2$</td>
</tr>
<tr>
<td>A1.</td>
<td>$I \rightarrow T$</td>
<td>$ma_1 = \text{Sig}_I(\text{aborted}, me_1)$</td>
</tr>
<tr>
<td>A2.</td>
<td>$T \rightarrow I$</td>
<td>$ma_2 = \text{Sig}_T(\text{aborted}, ma_1)$</td>
</tr>
<tr>
<td>R1.</td>
<td>$R \rightarrow T$</td>
<td>$mr_1 = \langle me_1, me_2 \rangle$</td>
</tr>
<tr>
<td>R2.</td>
<td>$T \rightarrow R$</td>
<td>$mr_2 = \text{Sig}_T(\text{aborted}, ma_1)$</td>
</tr>
<tr>
<td>E3.</td>
<td>$I \rightarrow R$</td>
<td>$N_I$</td>
</tr>
<tr>
<td>E4.</td>
<td>$R \rightarrow I$</td>
<td>$N_R'$</td>
</tr>
</tbody>
</table>

*R* obtains an *abort token* relative to the secret commitment $N_I$ and eventually the *intruder*, playing the originator, obtains a standard contract relative to the same contractual text and secret commitment.
SATMC finds the following attack:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$. $I \rightarrow R$ :</td>
<td>$me_1$</td>
<td></td>
</tr>
<tr>
<td>$E_2$. $R \rightarrow I$ :</td>
<td>$me_2$</td>
<td></td>
</tr>
<tr>
<td>$A_1$. $I \rightarrow T$ :</td>
<td>$ma_1 = \operatorname{Sig}_I(\text{aborted}, me_1)$</td>
<td></td>
</tr>
<tr>
<td>$A_2$. $T \rightarrow I$ :</td>
<td>$ma_2 = \operatorname{Sig}_T(\text{aborted}, ma_1)$</td>
<td></td>
</tr>
<tr>
<td>$R_1$. $R \rightarrow T$ :</td>
<td>$mr_1 = \langle me_1, me_2 \rangle$</td>
<td></td>
</tr>
<tr>
<td>$R_2$. $T \rightarrow R$ :</td>
<td>$mr_2 = \operatorname{Sig}_T(\text{aborted}, ma_1)$</td>
<td></td>
</tr>
<tr>
<td>$E_1$. $I \rightarrow R$ :</td>
<td>$me_1$</td>
<td></td>
</tr>
<tr>
<td>$E_2$. $R \rightarrow I$ :</td>
<td>$me'_2$</td>
<td></td>
</tr>
<tr>
<td>$E_3$. $I \rightarrow R$ :</td>
<td>$N_I$</td>
<td></td>
</tr>
<tr>
<td>$E_4$. $R \rightarrow I$ :</td>
<td>$N'_R$</td>
<td></td>
</tr>
</tbody>
</table>

$R$ obtains an abort token relative to the secret commitment $N_I$ and eventually the intruder, playing the originator, obtains a standard contract relative to the same contractual text and secret commitment.
SATMC finds the following attack:

\[
\begin{align*}
E1. & \quad I \rightarrow R : \quad me_1 \\
E2. & \quad R \rightarrow I : \quad me_2 \\
A1. & \quad I \rightarrow T : \quad ma_1 = \text{Sig}_I(aborted, me_1) \\
A2. & \quad T \rightarrow I : \quad ma_2 = \text{Sig}_T(aborted, ma_1) \\
R1. & \quad R \rightarrow T : \quad mr_1 = \langle me_1, me_2 \rangle \\
R2. & \quad T \rightarrow R : \quad mr_2 = \text{Sig}_T(aborted, ma_1) \\
E1. & \quad I \rightarrow R : \quad me_1 // \\
E2. & \quad R \rightarrow I : \quad me'_2 // \quad R \text{ has the} \\
E3. & \quad I \rightarrow R : \quad N_I // \quad \text{same contract} \\
E4. & \quad R \rightarrow I : \quad N'_R //
\end{align*}
\]

*R* obtains an abort token relative to the secret commitment *N* and eventually the *intruder*, playing the originator, obtains a standard contract relative to the same contractual text and secret commitment.
Again the problem lies in the formulation of the security property and not in the protocol.

In fact $R$ eventually obtains the same standard contract obtained by the intruder.

We rectify the problem by reformulating the goal as follows:

(B”) If an honest principal, say $A$, obtains an abort token, then any other principal has already a corresponding valid contract or will not be able to obtain one in the future, or $A$ already owns or eventually will obtain a corresponding valid contract.

No attack is found by SATMC while checking the protocol w.r.t. (B”).
Outline

1. Introduction
2. LTL Model Checking for Security Protocols
3. A Motivating Example: the ASW Protocol
   - The Protocol
   - The Assumptions on the Channels and on the Honest Principals
   - Security Properties
4. Modelling, Specification and Analysis using Set-rewriting and LTL
   - Specifying the Protocol Scenario and the Intruder
   - Specifying the Assumptions on the Channels and on the Honest Principals
   - Specifying Security Properties
5. Conclusions
Conclusions

- We have presented a **general framework** for security protocols (set-rewriting + LTL) that allows for the specification of:
  - assumptions on principals and channels
  - complex security properties

  that are normally not handled by state-of-the-art analysers.

- We have demonstrated the **effectiveness** of the approach. By analysing the **ASW protocol** we have:
  - found all known attacks + a new attack on the “patched” version
  - refined/revised the specification of fair exchange.

- Our analysis shows that the use of LTL
  - is **not only** useful for specifying and supporting the mechanical verification of security protocols,
  - but also **to support** the understanding and the specification of their security requirements.
Conclusions

- We have presented a general framework for security protocols (set-rewriting + LTL) that allows for the specification of:
  - assumptions on principals and channels
  - complex security properties

  that are normally not handled by state-of-the-art analysers.

- We have demonstrated the effectiveness of the approach. By analysing the ASW protocol we have:
  - found all known attacks + a new attack on the “patched” version
  - refined/revised the specification of fair exchange.

- Our analysis shows that the use of LTL
  - is not only useful for specifying and supporting the mechanical verification of security protocols,
  - but also to support the understanding and the specification of their security requirements.
Conclusions

- We have presented a general framework for security protocols (set-rewriting + LTL) that allows for the specification of:
  - assumptions on principals and channels
  - complex security properties

That are normally not handled by state-of-the-art analysers.

- We have demonstrated the effectiveness of the approach. By analysing the ASW protocol we have:
  - found all known attacks + a new attack on the “patched” version
  - refined/revised the specification of fair exchange.

- Our analysis shows that the use of LTL
  - is not only useful for specifying and supporting the mechanical verification of security protocols,
  - but also to support the understanding and the specification of their security requirements.