Automated Reasoning for Security Protocol Analysis

The ASW Protocol Revisited: A Unified View

Paul Hankes Drielsma and Sebastian Mödersheim
Information Security, ETH Zurich
Introduction

• ASW: an asynchronous, optimistic fair exchange protocol introduced by [Asokan, Shoup, Waidner].
  – Such protocols and their objectives are often beyond the scope of existing protocol analysis tools.

• We revisit the analysis of ASW:
  – We adopt a simple, unified view of the protocol that enables us to reason about protocol objectives.
  – We perform an automated analysis for both finite and infinite protocol sessions using two tools, OFMC and OFMC-FP
Protocol Objectives

• **Fair exchange**: At the end of a protocol execution, either both parties possess valid contracts, or neither does.

• **Effectiveness**: If two honest agents complete a protocol run and neither chooses to abort it, then both possess a valid contract.

• **Timely completion**: Both originator and responder can be sure of completion within a finite amount of time.

• **Non-repudiability**: A contract contains implicit proof of the agents’ acceptance of the contractual text.

• **Abuse-Freeness**: Neither party can prove to an outside verifier that he has the power to decide the outcome of the protocol.
The ASW Protocol (1/3)

Exchange subprotocol:
1. $O \to R : \quad me_1 = Sig_O(V_O, V_R, T, text, h(N_O))$
2. $R \to O : \quad me_2 = Sig_R(me_1, h(N_R))$
3. $O \to R : \quad N_O$
4. $R \to O : \quad N_R$

- Two rounds: exchange of public commitments followed by exchange of secret commitments
- Upon successful completion, both parties will be in possession of a standard valid contract of the form $me_1, me_2, N_O, N_R$. 
The ASW Protocol (2/3)

Abort subprotocol:
1. $O \rightarrow T : \quad ma_1 = Sig_O(\text{aborted, } me_1)$
2. $T \rightarrow O : \quad ma_2 = \begin{cases} 
\text{if } \text{resolved}(me_1) \text{ then } Sig_T(me_1, me_2) \\
\text{else } Sig_T(\text{aborted, } ma_1) \ ; \ \text{aborted}(me_1) = \text{true}
\end{cases}$

- If $O$ does not receive $R$’s reply $me_2$ “in time”, he may initiate the abort subprotocol with the T3P.

- T3P responds with an abort token if $me_1$ has not been previously resolved. Otherwise, he issues a replacement contract of the form $Sig_T(me_1, me_2)$ and marks $me_1$ as aborted.

- There are thus two forms of valid contract: standard and replacement.

- Note that an abort token is not proof that the associated contract is invalid. It merely asserts that the T3P has not and will not issue a replacement contract.
The ASW Protocol (3/3)

Resolve subprotocol:

1. $O \rightarrow T : \ mr_1 = \ me_1, me_2$
2. $T \rightarrow O : \ mr_2 = \begin{cases} \text{if } \text{aborted}(me_1) \text{ then } \text{Sig}_T(\text{aborted}, ma_1) \\ \text{else } \text{Sig}_T(me_1, me_2) ; \text{ resolved}(me_1) = \text{true} \end{cases}$

- Can be initiated by either $O$ or $R$ if the secret commitment expected is not received in time.

- Analogous to the Abort subprotocol: if $me_1$ has previously been aborted, the T3P responds with an abort token. Otherwise, he sends a replacement contract and marks $me_1$ as resolved.
The Unified View (1/3)

- We wish to view and reason about the protocol as a single, unified protocol with alternate execution paths. We view the abort and resolve subprotocols as part of the main exchange protocol.

- For instance, the unified originator role is as follows:

\[
\begin{align*}
\text{exchange}_1. \; O \rightarrow R : & \; me_1 \\
\text{if} \; \text{timeout} \; \text{then} & \; \text{abort}_1. \; O \rightarrow T : ma_1 \\
& \; \text{abort}_2. \; T \rightarrow O : ma_2 \; (\text{abort \; token \; or \; replacement \; contract}) \\
\text{else} & \\
\text{exchange}_2. \; R \rightarrow O : & \; me_2 \\
\text{exchange}_3. \; O \rightarrow R : & \; NO \\
\text{if} \; \text{timeout} \; \text{then} & \; \text{resolve}_1. \; O \rightarrow T : mr_1 \\
& \; \text{resolve}_2. \; T \rightarrow O : mr_2 \; (\text{abort \; token \; or \; replacement \; contract}) \\
\text{else} & \\
\text{exchange}_4. \; R \rightarrow O : & \; NR
\end{align*}
\]
The Unified View (2/3)

- This unified view yields an intuitive agent model. The internal states of an agent playing in the originator role are shown here.
The Unified View (3/3)

- Two fairness constraints: (a) timeout; (b) guaranteed response from the T3P ensure that any honest originator will eventually reach one of the final states.
Reasoning about the Unified View (1/2)

• We wish show that if an honest agent receives an abort token, then no other agent can obtain a valid contract.

• A simple meta-argumentation allows us to formulate protocol objectives as state-reachability problems in an infinite state transition system without fairness constraints:
  – We can ignore intermediate states.
  – We can therefore spare ourselves liveness considerations, e.g. “an agent can eventually reach a certain state”.
  – Rather, we check that if an agent reaches his final state, then his interests are ensured.
Reasoning about the Unified View (2/2)

- Like [Shmatikov & Mitchell] and others, we thus encode the protocol objectives as safety properties in a transition system without fairness constraints.

- Note that fairness constraints exclude traces; this is therefore a sound abstraction to make.

- The challenge is to find appropriate safety properties.
Encoding the Protocol Objectives

- Certain objectives (e.g. timeliness) can be shown to hold via simple reasoning about the protocol based on the unified view.

- In our analysis, we focus on the following aspect of fair exchange:

  If an honest agent receives an abort token, then nobody (except the T3P) can ever obtain a valid standard or replacement contract.

- This is a standard secrecy property within the scope of most protocol analysis tools.

- We note that we can check that this property is ensured even in sessions with the intruder.
An Attack on This Formulation of Fair Exchange

\( e_1. \ I \rightarrow R: \ me_1 \)
\( e_2. \ R \rightarrow I: \ me_2 \)
\( e_3. \ I \rightarrow R: \ NI \)
\( e_4. \ R \rightarrow I: \ NR \)

\( e_1'. \ I \rightarrow R: \ me_1 \)
\( e_2'. \ R \rightarrow I: \ me_2' \)

Intruder stops communication

\( a_1. \ I \rightarrow T: \ ma_1 \)
\( a_2. \ T \rightarrow I: \ abort \ token \)

\( r_1. \ R \rightarrow T: \ \{me_1, me_2'\} \)
\( r_2. \ T \rightarrow R: \ abort \ token \)

- OFMC reports the attack shown here, in which it is indeed the case that an honest \( R \) receives only an abort token, while the intruder receives a valid contract. Note, however, that \( R \) also possesses this contract, but received it in a different session.

- A questionable attack, but shows a subtlety of the objectives.
Conclusion

- Using OFMC-FP, we have verified, for infinitely many sessions, that the protocol fulfills a slightly weakened fair exchange objective.

- The unified view gives us a strong basis for reasoning about the protocol.

- This reasoning allows us to reduce several of the protocol’s objectives to standard secrecy and authentication goals digestible by standard analysis tools.

- Even with these simplified objectives, their modelling presents several practical challenges.