Probabilistic Analysis and Verification of the ASW Protocol Using PRISM

Salekul Islam[1] and Mohammad Abu Zaid[2]
(Corresponding author: Salekul Islam)

Department of Computer Science and Software Engineering, Concordia University[1]
1455 De Maisonneuve Blvd. West, Montreal, Quebec, Canada, H3G 1M8
Concordia Institute for Information Systems Engineering, Concordia University[2]
1455 De Maisonneuve Blvd. West, Montreal, Quebec, Canada, H3G 1M8
(Email: salek_is@cse.concordia.ca, Email: abuza_m@encs.concordia.ca)

(Received Jan. 5, 2007; revised June 13, 2007; and accepted Mar. 3, 2008)

Abstract

The ASW protocol is one of the prominent optimistic fair exchange protocols that is used for contract signing between two participants, the originator and the responder, with the aid of a trusted third party in case of a dispute. In this paper, the key security objectives of ASW protocol — fairness, effectiveness and timeliness — have been verified using a probabilistic model checking tool, PRISM. First, the security objectives of ASW protocol have been defined with probabilistic equations. The roles of the participants (i.e., the originator and the responder) and the trusted third party have been modeled in PRISM code. The security objectives of ASW protocol have been expressed using a temporal logic, PCTL. The PCTL expressions are analogous to the probabilistic equations that we have developed to define the security objectives. Next, the model is analyzed using these PCTL expressions, and different outputs have been observed. The outputs confirm the fairness of the ASW protocol. Moreover, the effectiveness and the timeliness of the protocol are also established. Hence, the key security properties of ASW protocol have been verified.

Keywords: Contract signing protocol, e-commerce protocol, fair exchange, probabilistic analysis

1 Introduction

In the last few years, the use of Internet and its diversity have increased tremendously. The end users' activities are no more restricted to browsing different sites, exchanging emails and communicating through text chat. Online businesses are being accomplished today by means of computers Internet connection. The rapid growth of e-commerce has inspired researchers in dealing with fair exchange protocols. Multiple parties, while using such a protocol, exchange information for different types of applications, such as contract signing, purchasing and certified mail.

A contract signing protocol allows a set of participants to exchange messages with each other in order to arrive in a final state in which each of them has a pre-agreed contract text signed by all other participants. A contract signing protocol is deemed to be *fair*, if the parties, which are involved in the exchange finish the protocol in a fair state. Thus, either all parties will get the non-repudiation evidence that the messages have been exchanged, or none of the parties gets anything valuable. A contract signing protocol is not difficult to design if the involved parties sit together and meet face to face. However, in a distributed environment such as the Internet, where the parties do not physically meet and trust each other, it is very difficult to design a fair contract signing protocol. In general, a fair contract signing protocol has several security objectives: fairness, timeliness, effectiveness, non-reputability, etc. Fairness is the most important security property, but it is also very difficult to model formally.

A number of contract signing protocols have been designed with different approaches in the last few years, and a category of optimistic fair exchange is introduced. The term *optimistic* comes from the nature of these types of protocols that require the involvement of a trusted third party only if something goes wrong. The third party must be trusted by all the contract signers. The trusted third party will act upon the reception of a request from one of the participants to clarify the situation regarding a dispute. Among the optimistic fair exchange protocols, the ASW [1] is the most prominent one. In this paper, we have verified the security objectives of ASW protocol using a tool, Probabilistic Symbolic Model Checker (PRISM) [11], which is designed to analyze the probabilistic behavior of a system. First, we have defined the security objectives — fairness and effectiveness — of ASW protocol with mathematical probabilistic equations. Next, the roles of the participants (i.e., the originator and the responder) and the trusted third party have been modeled in PRISM code. The security objectives of ASW protocol have been expressed using a temporal logic expressions, Probabilistic Computation Tree Logic (PCTL). The PCTL expressions are analogous to the probabilistic equations that we have developed to define the security objectives. Next, the model is analyzed using these PCTL expressions and different outputs are observed. Finally, the outputs verify the fairness, effectiveness and timeliness of the ASW protocol.

The rest of the paper is organized as follows: Section 2 will explain the ASW protocol and its security objectives, and Section 3 will discuss the related work that attempts to analyze the ASW protocol. Section 4 will present the roles of the involved parties using state diagram, the PRISM model we have developed. Section 5 will present the results we have found and the verification using the results. Finally, Section 6 will be the conclusion and outline the work that we have planned to accomplish in future.

2 The ASW Protocol

The ASW [1] is an optimistic fair exchange protocol for contract signing through which two participants, the originator, O and the responder, R make a commitment to a previously agreed contractual text. In this protocol, the trusted party, T will be communicated only if a dispute is occurred. T will react either by issuing a replacement contract, or by alerting the requester (i.e., O or R) that a replacement contract cannot be issued and the contract must be eventually terminated. One of the strong requirements of the ASW protocol is that the communication channels between the parties involved in contract signing and T must be secured. Moreover, the intruder, if he/she exists, cannot delay or block a message from reaching T forever. Hence, an honest participant can always communicate with T to obtain a replacement contract in case the expected message from the other party is delayed or lost due to a channel error.

2.1 Protocol Description

The ASW protocol has three sub-protocols: exchange, abort and resolve [8]. In the usual operation of the protocol, only the exchange sub-protocol is executed. The other two are initiated only if one of the participants decides to forcibly complete the protocol by involving T.

Exchange sub-protocol:

1) $O \rightarrow R : me_1 = Sig_O(V_O; V_R; T; text; h(N_O))$.

2) $R \rightarrow O : me_2 = Sig_R(me_1; h(N_R))$.

3) $O \rightarrow R : N_O$.

4) $R \rightarrow O : N_R$.

The exchange sub-protocol runs between O and R, and consists of four messages, which are expressed in BAN [4] notation. If both participants are honest and there are no network failures or intruder intervention, after execution of the exchange sub-protocol, both will be in possession of the same valid contract. During the exchange sub-protocol run, both O and R generate their own nonce, N_O and N_R respectively. A nonce represents the secret commitment of the party who has generated it. Thus, the parties compute their so-called public commitments by hashing these values, yielding $h(N_O)$ and $h(N_R)$. The exchange sub-protocol has two rounds. In the first round, each party expresses his/her public commitment but does not disclose his/her secret commitment. In the second round, they exchange their respective secret commitments. At this level, each party can hash this later and thus verify that the purported secret commitment he/she has received earlier corresponds to the public commitment of the first round. At the end of the second round, each party is in possession of a valid standard contract of the form $me_1;me_2;\ N_O;N_R$.

Abort sub-protocol:

1) $O \rightarrow T : ma_1 = Sig_O(aborted; me_1)$.

2) $T \rightarrow O : ma_2 =$
   `if` $resolved(me_1)$ `then` $Sig_T(me_1; me_2)$
   `else` $Sig_T(aborted; me_1); aborted(me_1) = true$.

If O does not receive R's reply, me_2 within an acceptable time frame, O may abort the protocol by invoking T. O sends a signed abort request, ma_1 indicating that he/she wishes to abort the protocol. T is assumed to maintain a permanent database of the contracts for which it has been called upon to arbitrate. If T has already asserted the validity of the contract (indicated by $resolved(me_1)$), then it sends O a replacement contract of the form $Sig_T(me_1; me_2)$. Otherwise, T replies with an abort token and adds an entry in its database of aborted contracts. Such a token does not render an existing contract invalid but rather serves as a promise from T that it has not previously resolved the contract in question and will not do so in the future.

Resolve sub-protocol:

1) $O \rightarrow T : mr_1 = me_1, me_2$.

2) $T \rightarrow O : mr_2 =$
   `if` $aborted(me_1)$ `then` $Sig_T(aborted; ma_1)$
   `else` $Sig_T(me_1, me_2); resolved(me_1) = true$.

The resolve sub-protocol is analogous to the abort sub-protocol but can be invoked by both participants. The parties will request resolution of a contract from T if they do not receive the secret commitment or nonce of the other party within a reasonable amount of time. A resolution request includes both messages (me_1 and me_2) from the first round of the exchange sub-protocol. If T

has already issued an abort token for the contract in question (i.e., $aborted(me_1)$ is $true$), T replies with an abort token. Otherwise, T issues a replacement contract and updates its own database that it has resolved the dispute of the contract in question.

2.2 Security Objectives

In the ASW protocol [1], the authors have identified four security objectives for an optimistic fair exchange. In the following, the first four security goals have been redefined from [1], and the last one, accountability of T is first introduced in [12]:

1) **Fairness:** Fair exchange implies that no single party enrolled in the contract signing process will gain privilege over the other. Two notions of fairness have been identified in [1]. In strong fairness, when the protocol is completed either both O and R will be in possession of valid contracts, or neither will receive any important information (e.g., identity) about the other party. In weak fairness, a party can prove to an arbiter that the other party has received or can still receive information without any further intervention from him/her. The requirement, that this protocol must fulfill is if one of the participants ends up with an abort token the other must not be in possession of a valid contract.

   Definition 1. *If $O_{VR} = O$ has a valid or replacement contract and $R_A = R$ has received abort token from T, then $Pr(O_{VR} \bigcap R_A)$ represents the probability that O has received a valid contract while R has not received any important information. If the protocol is strongly fair, O_{VR} and R_A will be mutually exclusive and $O_{VR} \bigcap R_A = \phi$. Therefore,*

   $$Pr(O_{VR} \bigcap R_A) = 0. \tag{1}$$

   Similarly, in case of strongly fair protocol, the probability of R has a valid or replacement contract while O has received an abort token will be zero. Therefore,

   $$Pr(R_{VR} \bigcap O_A) = 0. \tag{2}$$

2) **Effectiveness:** If the two parties (O and R) want to sign a contract while both of them are honest participants and none of them have chosen to abandon the current protocol run, then both of them must have a valid contract when the protocol is completed.

   Definition 2. *If $O_V = O$ has a valid contract and $R_V = R$ has a valid contract then $Pr(O_V|R_V)$ represents the probability that if R has a valid contract then O must have a valid contract. A contract signing protocol that holds effectiveness, must satisfy the following equation:*

   $$Pr(O_V|R_V) = 1. \tag{3}$$

   Similarly, the probability of if O has a valid contract then R must have a valid contract will be one. Therefore,

   $$Pr(R_V|O_V) = 1. \tag{4}$$

3) **Timeliness:** This is a guarantee of the completion of the protocol run, more specifically, O and R can be sure of the completion of the protocol within a finite amount of time. The specific amount of the completion time is to be agreed upon by the two parties before execution of the protocol.

4) **Non-repudiation:** The objective of non-repudiation is that the contract must hold an implicit proof of both parties' involvement. Commitment to the textual context of the contract is divided into two parts: non-repudiation of O and non-repudiation of R. In other words, O and R are fully responsible for their commitment, and none of them can deny what has been agreed on previously.

5) **Accountability of T:** If T is corrupted, or has chosen to behave in such a way to compromise fairness of the exchange, then this corrupt behavior can be proven to an external verifier.

3 Related Work

The need for safe, secure, and fair exchange protocols attracted researchers to explore this area. Several fair exchange protocols have been proposed and some studies have been done to verify the security properties of such protocols. The analyses of the ASW and the GJM protocols have been carried out with close attention. Most of the existing studies that analyze the ASW protocol are focused on authentication and secrecy properties in the presence of the Dolev-Yao model [7]. These are carried out to investigate possible attacks and weaknesses, and to examine for properties that were not in the design goal of the original ASW protocol [1]. For example, abuse-freeness, which was not mandated in the original specification though ASW provides it.

In the most recent work [3], Backes et al. develop a method based on reasoning logic to prove properties of the ASW and the GJM protocols. They have provided a very strong detailed reasoning under the Dolev-Yao model. Despite the fact that their method offers certain advantages over other analysis techniques, we have still found this method is limited in directly addressing the security properties of a protocol. The method is not easy to adapt, and researchers have to develop a template for each protocol to analyze the targeted protocol. Moreover, modeling the protocol timeliness property is not possible with this logic. Considering human factors, there is always possibility of error in formalizing the protocol of interest. Thus, an automated analysis tool is still a preferable choice.

Zhou, Deng and Bao [15] analyze the ASW certified mail protocol for flaws and weaknesses. They have proposed a variant protocol to overcome the breaches they have found, and informally analyze the new one. However, their study goal was only to check mainly for authentication and secrecy flaws.

Drielsma and Mödersheim [8] have encoded the fairness property to safety property, while adopting "the unified view" to perform meta-reasoning of the ASW protocol. They have used a model checker, AVISPA [2] to verify the security objectives of the ASW protocol under the Dolev-Yao model. In their analysis, they have demonstrated the authentication failures that were reported by Shmatikov and Mitchell in [14]. In their approach, again the original ASW objectives are not really verified. Though we have also used the unified view approach in modeling the roles of the participants, we disagree with their assumption of reducing the fairness to safety property. Additionally, they have not verified for timeliness and effectiveness properties.

Shmatikov and Mitchell have performed sequential analysis targeting both the ASW and the GJM [9] protocols using a finite-state model-checker, Murϕ [12, 13, 14]. The principal idea is to reduce the problem of fairness property to safety property. Their judgment has been developed by having a comprehensive overview of the protocol as a whole. They have embraced a detailed study for message exchange between the participants, and they are the first (to our knowledge) to address the authentication problem in ASW protocol and suggest a solution to overcome this weakness by signing the nonces under the Dolev-Yao model assumptions. However, they have checked abuse-freeness for the ASW protocol, while ASW is not designed to ensure the abuse-freeness property.

Chadha, Kanovich and Scedrov [6] use inductive arguments to analyze GJM protocol for abuse-freeness. Kremer and Raskin [10] analyze for abuse-free contract signing by using game-theory concepts, and carry out the analysis using a temporal logic model checker, Mocha, under certain assumptions. Chadha et al. [5] also study contract signing protocols in a game-theory model while they use very strong arguments to prove the non-existence of certain conditions.

We can conclude from these existing studies that the focus is on authentication and secrecy properties. Despite the fact that some researchers have addressed the fairness property, they have accomplished this under certain assumptions.

4 Modeling of ASW Protocol Using PRISM

4.1 Modeling Roles of the Participants

To facilitate the analysis of the ASW protocol with PRISM, we have identified the roles of the two parties (O and R) and T. These roles are presented in state diagrams, where a state diagram corresponds to each participant's normal roles in the protocol run and roles in case something goes wrong. We have considered the behavioral role of each party for all types of alternate situations. We have presented the transition of the internal states of the participants depending on the message sequence of the ASW protocol. Hence, when we present a message exchange, we refer to the high level description of the message without going into the detailed structure of it. In the state diagrams, two-dimensional labeled arrows are used to represent the sending or receiving of messages to or from the participant labeled inside the arrow. An incoming arrow represents the receiving of a message and an outgoing arrow represents the sending of a message. In the diagrams, a message $Xi(y)$ represents the ith step of a sub-protocol starting with the letter "X" that carries the message y. For example, $A1(ma_1)$ represents the 1st step of the abort sub-protocol that carries the message, ma_1.

4.1.1 Role of the Originator

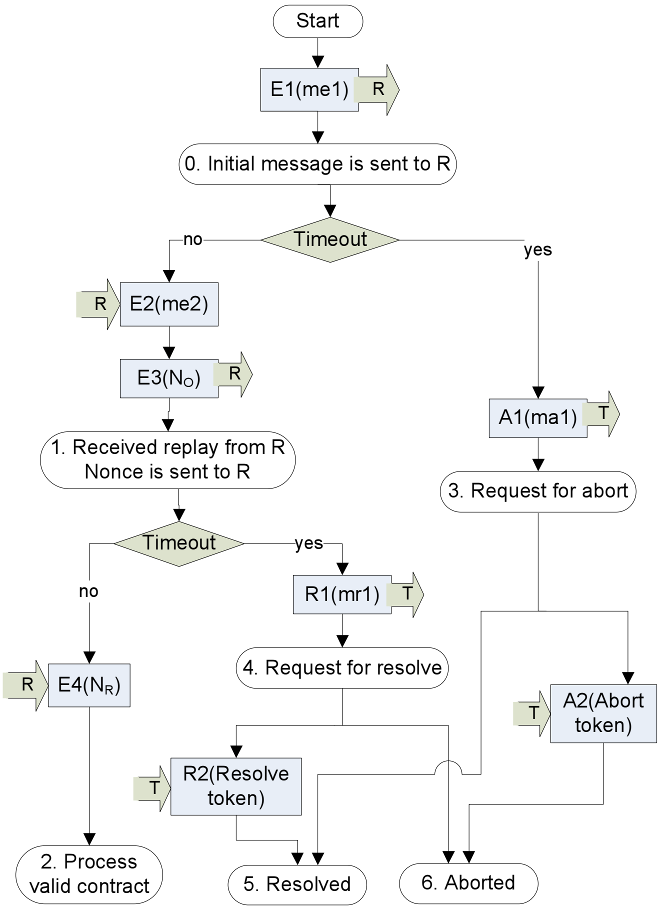

Figure 1: State diagram of the originator

The state diagram of O has been shown in Figure 1.

Zero, after sending the initial message, O waits for a reply from R until *timeout* occurs.

First, if O receives a reply before the *timeout* occurs, O sends his/her nonce, N_O and moves to a waiting state (similar to the Zero state).

Second, if O receives R's message before the *timeout* occurs confirming the reception of N_O, O reaches the final state with a valid contract.

Third, if the *timeout* occurs in the Zero state, O sends a message, ma_1 to T requesting it to abort the protocol. In this state, O waits for T's response without triggering the *timeout*, as the channels between the two parties (O and R) and T have been assumed to be secure and reliable in [1].

Fourth, if the *timeout* occurs in the First state, O sends a message, mr_1 to T requesting it to resolve the protocol. The waiting analogy is similar to the Third sate.

Fifth, O receives a resolve token containing a replacement contract from T.

Sixth, O receives an abort token from T.

4.1.2 Role of the Responder

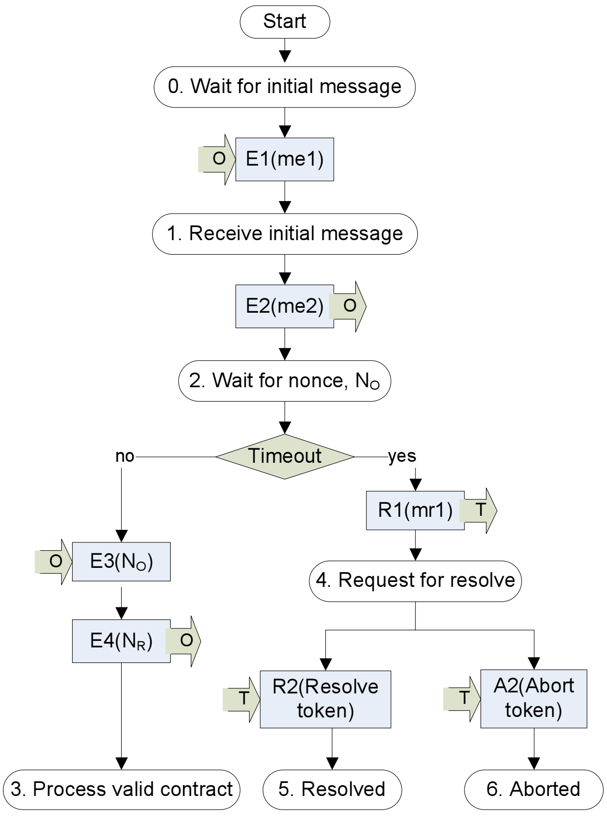

Figure 2: State diagram of the responder

Figure 2 shows the state diagram of R.

Zero, a waiting state for the initial message from O.

First, On receiving the initial message, me_1 from O, R being an honest participant sends the reply, me_2 to O and moves to a waiting state.

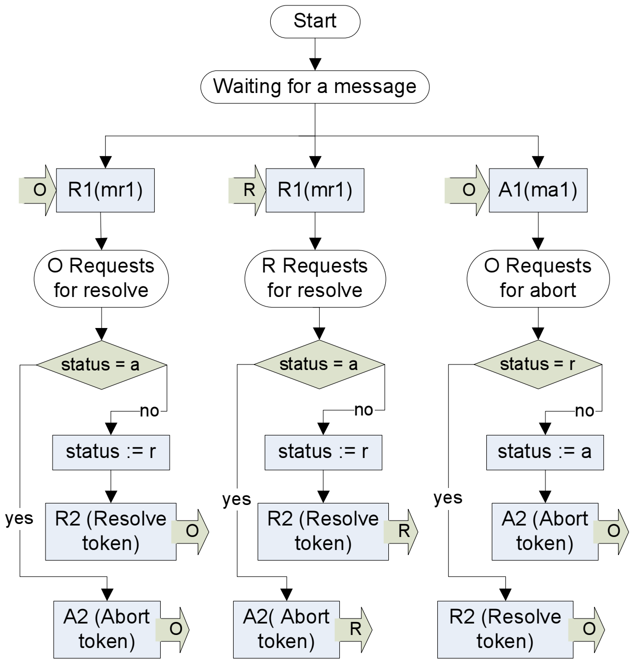

Figure 3: State diagram of the trusted third party

Second, R waits for a reply (the nonce, N_O) from O until *timeout* occurs.

Third, R receives the nonce, N_O from O, continues with normal protocol execution, sends his/her nonce, N_R, and moves to a final state with a valid contract.

Fourth, if *timeout* occurs in the Second state, R sends a message, mr_1 to T requesting it to resolve the protocol.

Fifth, R receives a resolve token containing a replacement contract from T.

Sixth, R receives an abort token from T.

4.1.3 Role of the Trusted Third Party

The state diagram of T is shown in Figure 3. T will be in a waiting state until one of the parties requests resolving or aborting the protocol. However, only O can request abort sub-protocol, whereas, both O and R can request resolve sub-protocol. T maintains a database, *status*, for each contract for which it has been called upon to arbitrate.

If T receives a request to resolve the protocol, it checks the value of *status*. If this value is equal to a, this protocol or contract has already been aborted, and an abort token will be sent to the requesting party. Otherwise, the *status* will be assigned r, and a resolve token will be sent to the requesting party.

If T receives a request to abort the protocol, it checks the value of *status*. If this value is equal to r, this protocol or contract has already been resolved, and a resolve token will be sent to O. Otherwise, the *status* will be assigned a, and an abort token will be sent to O.

4.2 PRISM

The PRISM (Probabilistic Symbolic Model Checker) [11] is a tool for modeling and analysis of probabilistic behaviors of a system. It is based on the construction of a precise mathematical model of a system, which is to be analyzed by the model checking tool. Properties of a system are expressed formally using a temporal logic, and analyzed against the constructed model. PRISM supports three types of probabilistic models: Discrete-Time Markov Chains (DTMCs), Markov Decision Processes (MDPs) and Continuous-Time Markov Chains (CTMCs). In practice, these models are specified by writing descriptions in PRISM language, a simple and high-level modeling language. For the specification of properties, the tool supports two types of temporal logics: Probabilistic Computation Tree Logic (PCTL) for DTMCs and MDPs models, and Continuous Stochastic Logic (CSL) for CTMCs models.

4.3 PRISM Model of ASW Protocol

We have developed the model of the ASW protocol by transforming the roles of different entities into PRISM language code. The model in PRISM language is shown in Figure 4. It has three modules: `parties` for the role of O and R, `t3p` for T and `timer` to present the occurance of *timeout*. The global variables `o` and `r` represent the states of O (see Figure 1) and R (see Figure 2) respectively. The variable `delay` (which gets any value from `1` to `10`) determines the occurrence of timeout. If this is less than `o_timeout` (`r_timeout`), O (R) will never be timed out. We have assigned the value of `o_timeout` and `r_timeout` during runtime.

5 Results Analysis and Verification

We have verified the three key objectives (i.e., fairness, effectiveness and timeliness) of the ASW protocol. The other two objectives (non-repudiation and accountability of T) have been addressed by the other researchers [5, 12, 13, 14]. According to Figure 1 and 2, there are two types of states for the role of O and R: final state and intermediate state. In a final state, a party is not waiting for any message, and either he/she has a valid contract or it has been aborted by T. In an intermediate state, he/she is waiting for a message from the other party. We have expressed different security objectives in terms of probability of reaching various final or intermediate states by using PCTL. However, PRISM always calculates either the maximum or the minimum probability of a PCTL property.

```
nondeterministic
const int o_timeout;             // timeout value of originator
const int r_timeout;             // timeout value of responder
global o : [0..6] init 0;
global r : [0..6] init 0;
global t_a : [0..1] init 0;    // 0=initial, 1=abort from origin
global t_r_o : [0..1] init 0;  // 0=initial, 1=resolve from origin
global t_r_r : [0..1] init 0;  // 0=initial, 1=resolve from resp

module parties
abort_to : [0..1];  // 0=no abort timeout, 1=abort timeout appears
resolve_to : [0..1]; // 0=no resolve timeout, 1=resolve timeout
     // flip coin any time after a message has been sent
  [] o=0 & r=0 -> (r'=1);
  [] o=0 & r=1 -> (r'=2);
  [] o=0 & r=2 & (delay<o_timeout) -> (o'=1);   // no timeout
  [] o=0 & r=2 & (delay>=o_timeout) -> (o'=3) & (t_a'=1)
     & (abort_to'=1); // Originator=3, run abort protocol
  [] o=3 & abort_to=1 -> (r'=4)&(t_r_r'=1);
                              // responder=4, run resolve protocol
  [] o=1 & r=2 & (delay<r_timeout) -> (r'=3);    // no timeout
  [] o=1 & r=2 & (delay>=r_timeout) -> (r'=4)&(t_r_r'=1);
                           // responder=4, run resolve protocol
  [] o=1 & r=3 & (delay<o_timeout) -> (o'=2);    // no timeout
  [] o=1 & r=3 & (delay>=o_timeout) -> (o'=4)&(t_r_o'=1)
           &(resolve_to'=1); // originator=4, run resolve protocol
endmodule

module t3p
status : [0..2] init 0;    // 0=undefined, 1=aborted, 2=resolved

  //abort subprotocol
  [] t_a=1 & status=2 -> (o'=5);         // originator=5, resolved
  [] t_a=1 & status!=2 -> (status'=1) & (o'=6);
                                          // originator=6, aborted
  // resolve subprotocol runs with originator
  [] t_r_o=1 & status=1 -> (o'=6);       // originator=6, aborted
  [] t_r_o=1 & status!=1 -> (status'=2) & (o'=5);
                                          //originator=5, resolved
  // resolve subprotocol runs with responder
  [] t_r_r=1 & status=1 -> (r'=6);        // responder=6, aborted
  [] t_r_r=1 & status!=1 -> (status'=2) & (r'=5);
                                          // responder=5, resolved
endmodule

module timer
delay : [0..10] init 0;
  [] delay=0 -> 1/10 : (delay'=1)
              + 1/10 : (delay'=2)
                .
                .
              + 1/10 : (delay'=10);
endmodule
```

Figure 4: PRISM code for ASW protocol

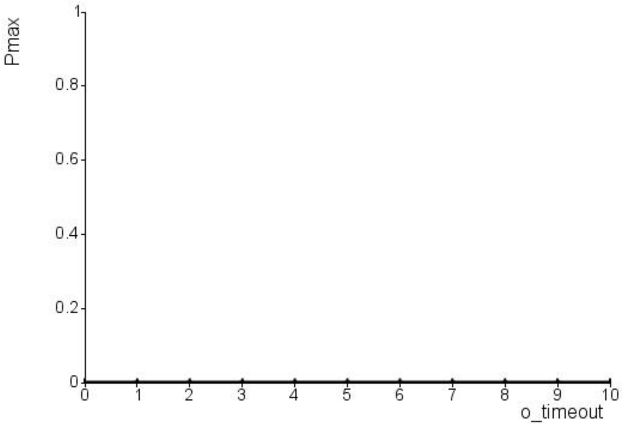

Figure 5: Maximum probability of a privileged responder

5.1 Fairness

For fairness, we have computed the maximum probability of reaching a state, where one of the participants has privilege over the other. There are two different cases. In case one, we have assumed a privileged responder while

R decides to misbehave or abort the protocol run, and O is in possession of a valid or replacement contract. The following PCTL property tests the condition, where O is in State 2 or 5 and R is in State 6:

$$Pmax = ?[trueU((o = 2)|(o = 5))\&(r = 6)] \quad (5)$$

From Figure 1 and Equation (1), we can deduce that `((o=2) | (o=5))` and O_{VR} express the same meaning and that is O is in possession of a valid or replacement contract. Further, from Figure 2 and Equation (1), we can deduce that `(r=6)` and R_A express the same meaning and that is R has aborted the protocol. Therefore, it is understandable that Equation (5) is analogous to Equation (1), and hence, the value of `Pmax` in Equation (5) must be equal to zero to satisfy the strong fairness of the ASW protocol.

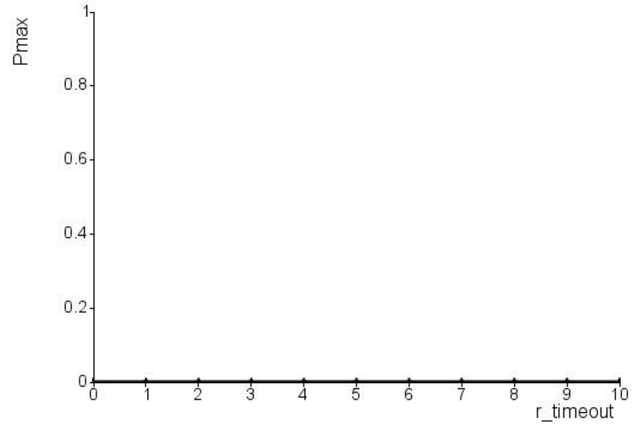

Figure 6: Maximum probability of a privileged originator

Figure 5 shows that the maximum probability, `Pmax` for R to have advantage over the originator is zero for all different values of `o_timeout`, which means this situation will never occur in real protocol run. This evidence proves that the protocol yields no privilege for R over O.

In case two, under the assumption of a privileged originator, O decides to misbehave or abort the protocol run, while on the other hand, R is in possession of a valid contract or a replacement contract. The following PCTL property expresses the condition, where R is in State 3 or 5, and O is in State 6:

$$Pmax = ?[trueU((r = 3)|(r = 5))\&(o = 6)]. \quad (6)$$

Following the logical reasoning of case one, from Figures 2 and 1, and from Equation (2), it can be inferred that Equation (2) is analogous to Equation (6), and hence, the value of `Pmax` in Equation (6) must be equal to zero to satisfy the strong fairness of the ASW protocol.

Figure 6 shows that the maximum probability, `Pmax` for O to have advantage over the responder is zero for all different values of `r_timeout`. The output establishes the fact that O will never gain any privilege over R during protocol run or contract signing.

From the above two results, it can be concluded that none of the parties is in an advantageous position compared to their counter part in the ASW contract signing protocol.

5.2 Effectiveness

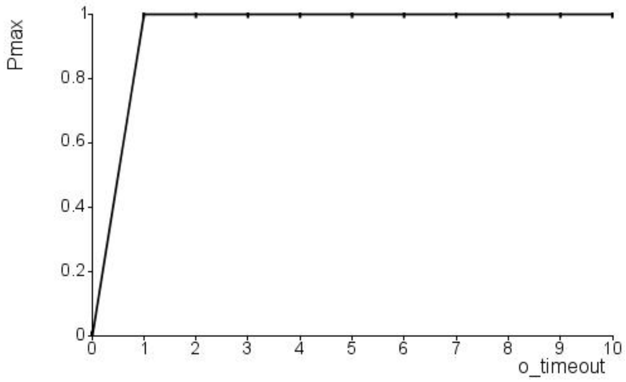

Figure 7: Maximum probability of both originator and responder in possession of a valid contract

To express effectiveness, we have tested the following PCTL property, where both of the parties are honest participants, have never intended to abandon the protocol, and finally have reached the states with the same valid contract (O in State 2 and R in State 3):

$$Pmax = ?[trueU(o = 2)\&(r = 3)]. \quad (7)$$

From Figures 1 and 2, and Equation (3), we can deduce that `(o=2)` and `(r=3)` express the same meaning as O_V and R_V do, respectively. Therefore, following the Equation (3), the value of of `Pmax` in Equation (7) must be equal to one to satisfy the effectiveness property of the ASW protocol.

The result we have obtained using PRISM is shown in Figure 7. It is demonstrated that the maximum probability, `Pmax` of effectiveness is always 100%, and change of the value of `o_timeout` has no effect (as long as `o_timeout` > 0) on `Pmax`. A similar result can be found for `r_timeout`. The derived results undoubtedly demonstrate the effectiveness of the ASW protocol.

5.3 Timeliness

Finally, to test timeliness, we have computed the maximum probability that one of the participants will be in a final state while the other is halted in an intermediate state of waiting. We have considered two cases. In the first one, R is waiting in one of the intermediate states (1, 2 or 4) and O has reached one of the final states (2, 5 or 6). The second case is opposite to the first one; O is waiting in one of the intermediate states (0, 3 or 4) and R has reached one of the final states (3, 5 or 6). These

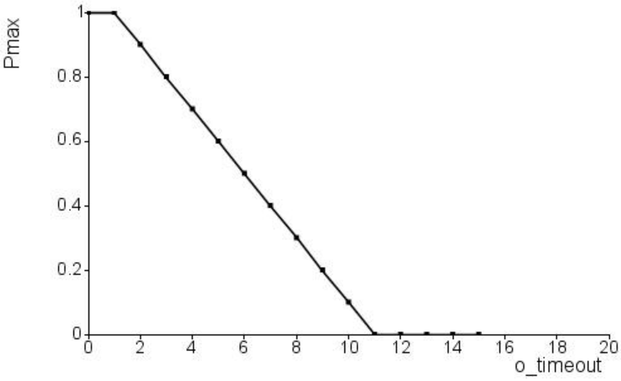

Figure 8: Maximum probability of responder in an intermediate state when originator has reached a final state

properties are interpreted by the following PCTL logics:

$$\begin{aligned} Pmax \; &=? \;\; [trueU((o=2)|(o=5)|(o=6)) \\ &\& \;\; ((r=1)|(r=2)|(r=4))]. \\ Pmax \; &=? \;\; [trueU((o=0)|(o=3)|(o=4)) \\ &\& \;\; ((r=3)|(r=5)|(r=6))]. \end{aligned}$$

We have obtained the same output for both scenarios, and the first one is shown in Figure 8. In this case, the value of r_timeout is fixed and is equal to 5, and the value of o_timeout varies from 0 to 15. The maximum probability, Pmax in Figure 8 drops with the increase of o_timeout. Eventually, it goes to zero when o_timeout becomes greater than delay. The results demonstrate the dependency of the outputs on the timeout values of the participants, and since these values are finite, it can be concluded that the ASW protocol satisfies the timeliness property.

6 Conclusion and Future Work

Our approach is the first to capture the security objectives without drifting from the original protocol specifications as stated in [1]. Hence, we have not focused on the weaknesses of authentication and secrecy properties. The contributions of the paper could be summarized as follows:

- The two major security objectives — fairness and effectiveness — of ASW protocol have been probabilistically defined with mathematical equations.

- The roles of the involved entities (i.e., the originator, the responder and the third party) of ASW protocol have been presented using state diagrams. Hence, the change of states due to the occurrence of any event are clearly shown.

- The first probabilistic model (to our knowledge) of ASW protocol has been developed using PRISM. The security objectives are also expressed using PCTL logical expressions.

- The key security objectives of ASW protocol have been successfully verified with the help of the PRISM model and the PCTL expressions. Thus, we have reasonably established that the security objectives of the ASW protocol conform to its requirements.

We believe a tool like PRISM will be very useful in automatic verification of e-commerce protocols in a probabilistic manner. In future, we intend to expand the study by exploring unbounded number of sessions, increasing the state spaces, and using PRISM to analyze other e-commerce protocols.

Acknowledgments

The authors would like to thank Dr. Mourad Debbabi, Professor of Concordia Institute for Information Systems Engineering for his valuable advices, keen interest and supervision of the project.

S. Islam acknowledges the support of Quebec Government through its FQRNT scholarship program and of Concordia University.

References

[1] N. Asokan, V. Shoup, and M. Waidner, “Asynchronous protocols for optimistic fair exchange,” *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp. 86-99, May 1998.

[2] *Automated Validation of Internet Security Protocols and Applications (AVISPA)*, http://www.avispa-project.org.

[3] M. Backes, et al., “Compositional analysis of contract signing protocols,” *Proceedings of the 18th IEEE Computer Security Foundations Workshop*, pp. 94-110, June 2005.

[4] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18-36, 1990.

[5] R. Chadha, J. Mitchell, A. Scedrov, and V. Shmatikov, “Contract signing, optimism and advantage,” *Journal of Logic and Algebraic Programming*, vol. 64, no. 2, pp. 189-218, 2005.

[6] R. Chadha, M. Kanovich, and A. Scedrov, “Inductive methods and contractsigning protocols,” *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 176-185, Nov. 2001.

[7] D. Dolev, and A. Yao, “On the security of public-key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198-208, 1983.

[8] P. H. Drielsma, and S. Mödersheim, “The ASW protocol revisited: A unified view,” *Electronic Notes in Theoretical Computer Science*, vol. 125, no. 1, pp. 145-161, 2005.

[9] J. A. Garay, M. Jakobsson, and P.D. MacKenzie, "Abuse-free optimistic contract signing," *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pp. 449-466, Jan. 1999.

[10] S. Kremer, and J. F. Raskin, "Game analysis of abuse-free contract signing," *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, pp. 206, June 2002.

[11] *Probabilistic Symbolic Model Checker (PRISM)*, http://www.cs.bham.ac.uk/∼dxp/prism/index.php.

[12] V. Shmatikov, and J. C. Mitchell, "Analysis of a fair exchange protocol," *Proceedings of the Workshop on Formal Methods and Security Protocols*, pp. 119-128, July 1999.

[13] V. Shmatikov, and J. C. Mitchell, "Analysis of abuse-free contract signing," *Proceedings of the 4th International Conference on Financial Cryptography*, pp. 174-191, Feb. 2000.

[14] V. Shmatikov, and J. C. Mitchell, "Finite-state analysis of two contract signing protocols," *Theoretical Computer Science*, vol. 283, no. 2, pp. 419-450, June 2002.

[15] J. Zhou, R. Deng, and F. Bao, "Some remarks on a fair exchange protocol," *Proceedings of the 3rd International Workshop on Practice and Theory in Public Key Cryptography*, pp. 46-57, Jan. 2000.

Salekul Islam has been enrolled in the Ph.D. program in Computer Science at Concordia University since 2004. He graduated from Bangladesh University of Engineering and Technology (BUET) in 2000 with a Bachelor of Science in Computer Science and Engineering and from Concordia University in 2003 with a Master of Computer Science. He was a Junior Lecturer in the School of Communication, Independent University Bangladesh (IUB), Dhaka, from September 2000 to April 2001. His research interests are in the design, analysis and validation of network protocols for secured multicast.

Mohammad Abu Zaid received the B.Sc. in ECE from An-Najah National University in 1998, and M. Eng. in Information Systems security from Concordia University in 2008. His research interests include information security, MANET security, and network security.