Research Journal of Applied Sciences, Engineering and Technology 6(9): 1538-1542, 2013
DOI:10.19026/rjaset.6.3866
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2013 Maxwell Scientific Publication Corp.
Submitted: November 08, 2012 Accepted: January 21, 2013 Published: July 15, 2013

**Research Article**

# Formal Analysis of a Fairness E-Commerce Protocol

[1, 2]Mei Zhang and [2]Jing-Hua Wen
[1]School of Accountancy,
[2]Information Institute, Gui Zhou University of Financial and Economics, Guiyang, China

**Abstract:** In this study, a new approach is proposed for analyzing non-repudiation and fairness of E-commerce protocols. The authentication E-mail protocol CMP1 is modeled as finite state machine and analyzed in two vital aspects-non-repudiation and fairness using SMV. As a result, the CMP1 protocol is not fair and so we have improved it. This result shows that it is effective to analyze and check the new features of E-commerce protocols using SMV model checker.

**Keywords:** E-commerce Protocols, fairness, symbolic model verification

## INTRODUCTION

The widely use of networks and E-commerce has brought great convenience to our daily life. The basis of normal e-commerce transaction between the seller and customers is the secure e-commerce protocols which must also include the two important properties, namely, non-repudiation and fairness (Deng and Gong, 1996), in addition to those basic properties such as secrecy, security, authentication and integrity. So there are special and higher demands of the performance and function of the e-commerce protocol. To attain this goal, the e-commerce protocols must be analyzed with special formal analyzing tools. BAN logic method, stand space method, process algebra, computing information theory method and Petri net method are some influential ones in the past years. In the middle 1990s Kailar noticed the importance of the formal analyzing of accountability and non-reputation in E-commerce protocols firstly and he advanced the famous Kailar logic. But Kailar logic has so many non-formal hypotheses and cannot be used to analyze fairness.

Model checking method check whether a finite state system meets its design specification automatically using the state space searching method (Marrero *et al*., 1997). The advantage of the method lies in its automatic checking process and high checking speed and efficiency and it can give the reason why a certain property doesn't meet the needs, according to which the system can be improved. The method has greatly developed since its birth. Model checking doesn't have those disadvantages existing in Karlar logic and has been used in the security protocol checking area gradually. But e-commerce protocol checking with model checking is staying in elementary level, mainly used in conventional properties checking such as security, authentication and integrity.

This study will model, analyze and check the new features of E-commerce, that is, no repudiation and fairness, using the model checking method. And via the actual analyze checking of the email protocol CMP1, we have found that this protocol doesn't meet the fairness requirement.

## CMP1 PROTOCOL

**CMP1 protocol description:** CMP1 protocol is the non-repudiation E-mail protocol advanced in reference (Deng and Gong, 1996), mainly running on the message processing system defined by X.400. This protocol provides non-reputation service for secure transfer of email recurring to the reliable third party TTP (Medvinsky and Neuman, 1993). The detailed formal description is as follows:
CMP1 protocol:

- $A \rightarrow B : h(m), \{k\}_{K_{TTP}}, \{\{m\}_{Ka^{-1}}\}_k$
- $B \rightarrow TTP : \{h(m)\}_{K_b^{-1}}, \{k\}_{K_{TTP}}, \{\{m\}_{Ka^{-1}}\}_k$
- $TTP \rightarrow B : \{\{m\}_{Ka^{-1}}\}_{K_{TTP}^{-1}}$
- $TTP \rightarrow A : \{\{h(m)\}_{K_b^{-1}}, (B,m)\}_{K_{TTP}^{-1}}$

where, A and B stands for the email sender and the email receiver, k for the conversation key, Ka, $Ka^{-1}$, Kb, $Kb^{-1}$, $K_{TTP}$, $K_{TTP}$, $K_{TTP}^{-1}$ for the public key and private key of the sender and the receiver and the reliable third party TTP, respectively. The running process is explained as follows:

**Corresponding Author:** Mei Zhang, School of Accountancy, Gui Zhou University of Financial and Economics, Guiyang, China

Firstly, the sender selects a conversation key k and then sends the summary h(m) of message m, underwritten message m encrypted using k (namely, $\{(m\}\ Ka^{-1}\}_k))$ and encrypted conversation key to receiver $B$.

Secondly, B underwrites h(m) and then sends it together with the latter two to TTP. When he receives what B has sent him, TTP gets the $\{m\}_{Ka}{}^{-1}$ through decryption.

Thirdly, TTP sends B his private key after underwriting.

And for the forth step, TTP sends the summary underwritten by B and (B, m) underwritten using his own private key to A.

**Non-repudiation and fairness:** The basis of normal e-commerce transaction between the seller and users is the secure E-commerce protocols which must also include the two important properties, namely, no-repudiation and fairness, in addition to those basic properties, for example, secrecy, security, authentication, integrity. This is because the dispute about transaction has become a general issue. In order to settle the dispute we need to use non-repudiation. At the same time, the fairness of the protocols is even more necessary. So fairness and non-repudiation should be the two important properties in transaction (Xie and Zhang, 2004). The fairness means that at any stage during the protocol's running any participant of the protocol won't dominates. And there exists the difference between the definitions of strong fairness and feeble fairness (Zhou and Gollman, 1996). The definition of fairness is as follows:

**Definition 1:** That a non-repudiation protocol is fair means that the protocol can provide the sender and the receiver respectively valid non-repudiation evidence when the a running transaction of the protocol ends and that at any stage which includes termination due to exception during the transaction, no participant dominates any other participants, namely, no participant can get his opponent's non-repudiation evidence when his opponent has not received any valid non-repudiation evidence.

The former part of the definition defines the non-repudiation of the protocol actually and the latter part defines the fairness for the sender and the receiver to get the opponent's non-repudiation evidence respectively (Mitechell *et al.*, 1997).

The CMP1 protocol has the non-repudiation after Kailar logic formal analyze, but Kailar logic cannot prove whether or not this protocol has fairness (Zhou *et al.*, 2001). The improved method by Qing Sihan and others has gotten rid of this disadvantage, but cannot do the automatic checking.

## SYMBOLIC MODEL VERIFIER TOOL SMV

SMV is the model checking tool software developed in 1992 by Dr. L. McMillian of CMU (SMV Introduce, 2004) . SMV is based on symbolic model verification technique, for which the SMV has become the software's name. SMV was an experiment tool for hardware checking in the early days in order to study the possibility of symbolic model checking application. SMV has become a popular tool for analyzing finite state concurrent system nowadays. SMV has his set of specification language for describing finite state concurrent system. In order to verify a system using SMV, that system must be described with the specification language, namely, to build the finite automaton model of every module of the system and the global state model which is the Kripke model and that system's properties which will need to be verified must denoted using CTL. Then for the next step the SMV will run and give the modeling result.

**Definition 2:** Kripke structure is a five-tuple: M = (S, Q, R, AP, L), in which $S$ is the finite set of states, $Q \subseteq S$ is the initial set, $R \subseteq S \times S$ is the transformation relation, AP is the set of the atomic statements and their denials, L： $S\text{->}2^{AP}$ is the label function which returns the set of all the true atomic statements in $s \in S$, which is a subset of the set AP of the atomic statements:

After it has received the input submitted by the user, SMV first picks up the migratory system expressed in OBDD from the system specifications and then searching algorithms based on OBDD are used to traverse the model defined by the system and check if the specification holds or not and whether the system meets the properties described by CTL and in the end give the true or false result. And when the final result is false SMV will send the counter example which causes the false result. The working principle of SMV is showed in Fig. 1.

SMV system is a tool which is described with sequential logic CTL used to checking finite state system. SMV input language can be used to describe finite state system which can range from asynchronous to synchronous and from specific to abstract. You can describe the system through synchronous incompact description or via abstract asynchronous network and uncertain process. The number of a system's states will increase greatly as its complexity increases, finally causing the states-explosion problem which restricts the further development of model checking technique. Many methods has been advanced to lighten the states-explosion problem. The symbolic model verifying tool SMV based on OBDD's searching algorithm has efficiently lighten the states-explosion problem. Now SMV can verify as many as $10^{130}$ states.

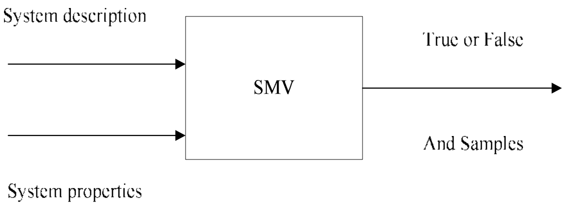

Fig. 1: SMV's working principle

## CMP1 PROTOCOL ANALYSES

CMP1 protocol includes the sender A, the receiver B and the reliable third party TTP as its principal communication parts (Cederquist *et al*., 2005). Only non-repudiation and fairness are considered, assuming the protocol is secure so the third party attacker model won't be included. But in order to be more actuarial, we need to consider the protocol runs in unreliable channels, that is, every step can be interrupted. We will use three FSM models to describe the three principal communication parts' activities respectively and the three FSM models stay in the same module.

**Description of the protocol message:** To describe messages in the CMP1 protocol, we construct the struct type message as shown in Fig. 2.

In which POO, POR are the non-repudiation evidences of the sender and the receiver respectively, message type the message type field, key the key field.

**Protocol's finite state system model:** Only non-repudiation and fairness are considered in this study, assuming the protocol is secure so the third party attacker model won't be included. That is to say, the set of principal parts in the protocol is {the sender $A$, the receiver $B$, the reliable third party $TTP$}. It is assumed that the reliable third party TTP is honest and fair and will perform the protocol strictly, but that A and B are not always honest and will perhaps interrupt the protocol on his own behalf (Xue and Feng, 2006). The three principal parts correspond to a FSM respectively in the SMV system. The three FSM are in the same module and every principal part is an instance of the module. The three instances corresponding to the three principal parts are Sender, Receiver and TTP Service respectively. Every principal part selects his corresponding automaton to run when the protocol is running. Their state transformations are listed in Fig. 3, 4 and 5, respectively (in which stands for sending message and stands for receiving message).

The sender A's states are {Start, A_G_m1, A_W_m2, A_W_m4, End}, in which W, G stands for waiting and generating, respectively, for example,

typedef message struct
{
message type: {Start, m1, m2, m3, m4, End};
key: { $k$ , $K_a$ , $K_b$ , $K_{b^{-1}}$ , $K_{a^{-1}}$ , $K_{TTP}$, $K_{TTP}{}^{-1}$ };
*POO*: $\{m\}_{K_a{}^{-1}}$ ;
*POR*: $\{\{h(m)\}_{K_b{}^{-1}},(b,m)\}_{K_{TTP}{}^{-1}}$ ;
}

Fig. 2: Defination of struct type message

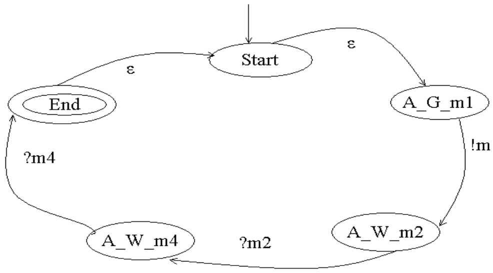

Fig. 3: The sender A's state transformation

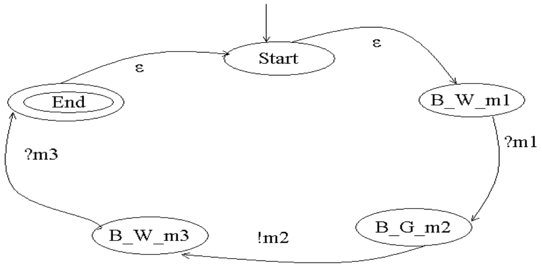

Fig. 4: The receiver B's state transformation

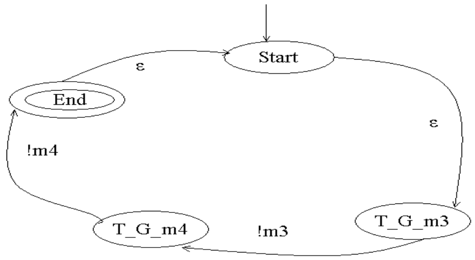

Fig. 5: The reliable third party TTP's state tranformation

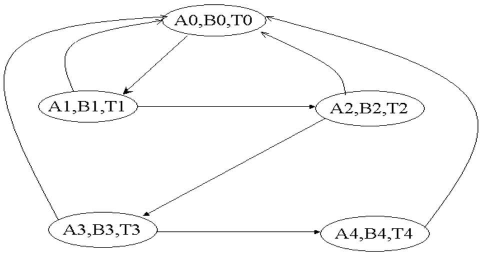

Fig. 6: The global state transformation (Kripke model)

A_G_m1 stands for the state that the sender A generates message 1. The sender B's states are {Start, B_W_m1, B_G_m2, B_W_m3, End}. And the reliable third party's states are {Start, TTP_G_m3, TTP_G_m4, End}. A's, B's and T's initial state are all Start. If it want to communicate with B, A enters A_G_m1 state automatically, selects responder B, generates message 1, sends message 1 to B and then enters A_W_m2 state and waits to receive message 2. When it has received message 2, A judges whether message 2 meets the protocol's requirement. If message 2 meets the requirement, A enters A_W_m4 state, waits to receive message4. When it has received message 4, A judges whether message 4 meets the protocol's requirement. If message 4 meets the requirement A will enter the End state and the current running of the protocol will ends. That A enters End state means that A has completed one message sending process. B's transformation is much the same as A's.

Synthesizing the three automata will get the whole system's global state transformation (Kripke model), which is in Fig. 6. Every step in the protocol is interruptible. Ai, Bi and Ti stands for the sender A's, the receiver B's and the reliable third party TTP's states, respectively, after the no i step in the protocol.

In the protocol modeling module, we use B_Rec_POO and A_Rec_POR to stand for whether the sender and the receiver has receive his opponent's non-repudiation evidence. The initial values for the two variables are all 0. When one non-repudiation evidence is received the corresponding value is added 1 to.

**Protocol properties CTL description:**

**Non-repudiation:** The sender's non-repudiation evidence and the receiver's non-repudiation evidence respectively are:

POO: $\{m\}_{K_a{}^{-1}}$

POR: $\{\{h(m)\}_{K_b{}^{-1}},(b,m)\}_{K_{TTP}{}^{-1}}$

The protocol meets the non-repudiation requirement namely: when the protocol ends, the sender receives POR and the receiver receives POO, describing using CTL as:

AF (Sender.A_Rec_POR>0)
AF (Receiver.B_Rec_POO >0)

**Fairness:** The protocol's fairness requires that whenever the protocol ends, the sender receives POR if and only if the receiver receives POO. The sender and the receiver are on an equality with each other, describing using CTL as:

AG (Sender.A_Rec_POR
Receiver.B_Rec_POO)

**Checking result analysis and improvement of the protocol:** We has input the protocol model above and the requirement for non-repudiation and fairness into the SMV and found that if the protocol model runs and ends normally the protocol model meets the non-repudiation requirement but won't meet the fairness requirement. After analysis we found that after the third step of the protocol has been executed, the receiver $B$ has received the senders' non-repudiation evidence POO. Because every step in the protocol can be interrupted, if the protocol interrupted right just after the third step's execution, the sender A will not be able to receive the receiver's the non-repudiation evidence POO. So the protocol is unfair to A. We has improved the protocol as follows:

- $A \to B: h(m),\{k\}_{K_{TTP}},\{\{m\}_{Ka^{-1}}\}_k$
- $B \to TTP: \{h(m)\}_{K_b{}^{-1}},\{k\}_{K_{TTP}},\{\{m\}_{Ka^{-1}}\}_k$
- $B \leftrightarrow TTP: \{\{m\}_{Ka^{-1}}\}_{K_{TTP}{}^{-1}}$
- $A \leftrightarrow TTP: \{\{h(m)\}_{K_b{}^{-1}},(B,m)\}_{K_{TTP}{}^{-1}}$

In the protocol above, we inherit the basic symbols in the CMP1 protocol. And according to Zhou-Gollman protocol thinking in Reference (Zhou and Gollman, 1996), $B \leftrightarrow TTP: m$ stands for B getting message m from the reliable center C via ftp operations for many times. The improved protocol above will make the two communication opponents be able to equally and forwardly get each other's non-repudiation evidence, which can be easily verified through the SMV model checking tool.

## CONCLUSION

This study has put forward method which analyze the CMP1 protocol from E-commerce's two aspects, namely, non-repudiation and fairness, adopting the model checking tool SMV and has found the disadvantages in the CMP1 protocol successfully. Compared to the successful logic analysis technique advanced in Reference (Zhou *et al.*, 2001) by Qing Sihan, model checking has additional advantages: automatically running and when the system model doesn't meet the system specification, the model checker will generate those counterexamples automatically. In our opinion, what is more important than the checking process and checking result in this study is that verifying the e-commerce protocol's special properties using SMV is an effective method.

## ACKNOWLEDMENT

Thank Project Supported by Basic Research Fund of Guizhou Provincial Science and Technology Department (NO: QianKeHeJZi [2011] 2191). Thank

Project Supported by Nomarch Fund of Guizhou Provincial excellence science and technology education person with ability ((2007)41). Thank Project Supported by Regional Science Fund of National Natural Science Foundation of China (Project approval number: 41261094.

## REFERENCES

Cederquist, J., R. Corin and M.T. Dashti, 2005. On the quest for impartiality: Design and analysis of a fair non-repudiation protocol. Proceedings of 2005 International Conference on Information and Communications Security, Beijing, China, pp: 7-39. Lecture Notes in Computer Science 3783, Springer-Verlag, ISBN: 3-540-30934-9.

Deng, R.H. and L. Gong, 1996. Practical protocols for certified electronic mail. J. Netw. Syst. Manag., 4(3): 279-297.

Marrero, W., E.M. Clarke and S. Jha, 1997. A Model checker for Authentication Protocols. Proceeding of (DIMACS) Workshop on Design and Formal Verification of Security Protocols, DIMACS Center, CoRE Building, Rutgers University, Piscataway, NJ, September 3-5, 1997.

Medvinsky, G. and B.C. Neuman, 1993. Netcash: A design of practical currency on the Internet [A]. Proceeding of the ACM Conference on Computer and Communication Security, ACM Press, New York, pp: 76-83.

Mitechell, J., M. Mitechell and U.S. Stern, 1997. Automated analysis of cryptographic protocols using mur. Proceeding of the IEEE Symposium on Security and Privacy, IEEE Computer Society Press, USA, pp: 141-151.

SMV Introduce, 2004. Retrieved from: http://www-cad.eecs.berkeley.edu/~kenmcmil.

Xie, X.Y. and H.G. Zhang, 2004. Fairness analysys of E-commerce protocols based on finate state model. Comput. Appl., 24(6): 38-44.

Xue, R. and D.G. Feng, 2006. The approaches and technologies for forma l verification of security protocols. Chinese J. Comput., 29(1): 1-20.

Zhou, J. and D. Gollman, 1996. A Fair Non-repudiation Protocol. In: Roscheisen, M. and C. Senban (Eds.), Proceeding of 1996 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, Oakland, California, pp: 55-61.

Zhou, D.C., S.H. Qing and Z.F. Zhou, 2001. A new approach for the analysis of electronic commerce protocols. J. Softw., 12(9): 1318-1328.