Symbolic Model Checking and Analysis for E-Commerce Protocol

WEN Jing-Hua1,2, ZHANG Mei1, LI Xiang2

(1. Information Institute, Guizhou College of Financial and Economics Guiyang 550004 China; 2. Institute of Software and Theory, Guizhou University Guiyang 550025 China)

Abstract 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 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.

Key words e-commerce protocols; fairness; symbolic model verification

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, in addition to those basic properties such as secrecy, security, authentication and integrity[1]. 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 (was published first by Burrows, Abadi and Needham), 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-repudiation 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 methods[2]. The advantages 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 Kailar 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 paper 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 analysis checking of the email protocol CMP1, we have found that this protocol doesn’t meet the fairness requirement.

1 CMP1 Protocol

1.1 CMP1 Protocol Description

CMP1(Certified mail protocol)protocol is the non-repudiation e-mail protocol advanced in Ref.[1], mainly running on the message processing system defined by X.400. This protocol provides non-repudiation service for secure transfer of email recurring to the reliable third party trusted third party (TTP)[3]. The detailed formal description of CMP1 protocol is as follows:

1) A → B: \( h(m), \{k\}_KTP, \{m\}_{K_A^{-1}} \)
2) B → TTP: \( h(m), \{k\}_KTP, \{m\}_{K_A^{-1}} \)
3) TTP → B: \( \{m\}_{K_A^{-1}}, \{k\}_KTP \)
4) TTP → A: \( \{h(m), \{k\}_KTP, \{B, m\}_KTP \} \)

where A and B stand for the email sender and the email receiver, \( k \) for the conversation key, \( K_A \), \( K_A^{-1} \), \( K_B \), \( K_B^{-1} \), \( K_TTP \), \( K_TTP^{-1} \) 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.

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, $\{ m \}_{k_x}$) 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 $\{ m \}_{k_x}$ 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 private key to A.

1.2 Non-Repudiation and Fairness

The basis of normal e-commerce transaction between the seller and customers is the secure e-commerce protocols which must include the two important properties: non-repudiation and fairness. 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\(^4\). 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\(^5\). 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\(^6\).

The CMP1 protocol has the non-repudiation after Kailar logic formal analysis, but Kailar logic cannot prove whether or not this protocol has fairness\(^7\). The improved method by Qing Sihan and others has got rid of this disadvantage, but cannot do the automatic checking.

2 Symbolic Model Verifier Tool SMV

SMV is the model checking tool software developed in 1992 by Dr. L. McMillian of Carnegie Mellon University\(^8\). SMV is based on symbolic model verification technique, for which the SMV has become the software’s name. SMV was an experimental 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. Then we specify system’s properties which will need to be verified must denoted using CTL\(^9\)\(^1\)\(^1\). Then for the next step the SMV will run and give the modeling result.

**Definition 2** Kripke structure is a five-tuples: $M=(S, Q, R, AP, L)$, in which $S$ is the finite set of states, $Q(\subseteq S)$ the initial set, $R(\subseteq S \times S)$ is the transformation relation, AP is the set of the atomic statements and their denials, $L: S \rightarrow 2AP$ 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 having received the input submitted by the user, SMV first picks up the migratory system expressed in ordered binary decision diagram (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 computation tree logic (CTL) and in the end give the true or false result. When the final result is false, SMV will the counterexample which cause the false result. The work flow of is shown as 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 ranges 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 when its complexity increases, finally causing the states-explosion problem which restricts the further development of model checking technique. Many methods has been advanced to solve the states-explosion problem. The symbolic model verifying tool SMV based on OBDD’s searching algorithm has efficiently solved the states-explosion problem. Now SMV can verify as many as $10^{130}$ states.

3 CMP1 Protocol Analysis

CMP1 protocol includes the sender A, the receiver B and the reliable third party TTP as its principal communication parts. 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 actual, 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.

3.1 Description of the Protocol Message

To describe messages in the CMP1 protocol, we construct the struct type message as follows:

```c
typedef message struct{
    messagetype: {Start, m1, m2, m3, m4, End};
    key: {k, K_A, K_B, K_tTP, K_tTP^-1};
    POO: {m} k_A^-1 ;
    POR: {h(m)} k_B^-1, (b, m) K_tTP^-1 ;
};
```

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

3.2 Protocol’s Finite State System Model

Only non-repudiation and fairness are considered in this paper, 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 they will perform the protocol strictly. However A and B are not always honest and will perhaps interrupt the protocol on his own behalf. 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.2, Fig.3 and Fig.4 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, 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 message 4. 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 ends. That A enters End state means that A has completed one message sending process. B's transformation is the same as A's. Synthesizing the three automata will get the whole system's global state transformation (Kripke model) as shown in Fig.5. Every step in the protocol is interruptable. A_i, B_i and T_i stand for the sender A's, the receiver B's and the reliable third party TTP's states, respectively, after the i-th step in the protocol.

3.3 Protocol Properties CTL Description

1) 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}, (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).

2) 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)

3.4 Checking Result Analysis and Improvement of the Protocol

We have inputted the protocol model above, 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 sender’s non-repudiation evidence POO. Since 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.

1) A \rightarrow B : h(m), \{k\}_{K_{TTP}}, \{m\}_{K_A}^{-1}, \{m\}_{K_A}^{-1}
2) B \rightarrow TTP : \{h(m)\}_{K_B}, \{k\}_{K_{TTP}}, \{m\}_{K_A}^{-1}
3) B \leftrightarrow TTP : \{m\}_{K_A}^{-1}, \{m\}_{K_A}^{-1}
4) A \leftrightarrow TTP : \{(h(m))_{K_B}, (B, m)\}_{K_{TTP}}^{-1}

In the protocol above, we inherit the basic symbols in the CMP1 protocol. According to Zhou-Gollman protocol thinking in Ref.[5], 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.

4 Conclusions

This paper has put forward a 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 with the successful logic analysis technique advanced in Ref.[7] 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 is to verify that the e-commerce protocol’s special properties using SMV is an effective method.

References


Brief Introduction to Author(s)

WEN Jing-hua (文静华) was born in Guizhou province, China, 1975. He graduated from North Jiaotong University in 1996. He obtained Master degree from GuiZhou University of Computer Science in 2002. Since 1999, he has been working in GuiZhou College of Financial and Economics as a lecturer. His research interests include computer software and information security.

ZHANG Mei (张 梅) was born in Guizhou province, China, 1974. She graduated from WuHan Mapping University of Science and Technology in 1996. She obtained Master degree from GuiZhou University of Computer Science in 2002. Since 1999, she has been working in GuiZhou College of Financial and Economics as a lecturer. Now she is engaged in the field of computer software and database technology.

LI Xiang (李祥) was born in Guizhou province, China, 1942. He graduated from Guizhou University in 1966. Since 1980, he has been working in GuiZhou University as a Professor. Now he is engaged in the field of cryptography and logic.