Formal Method for Security Analysis of Electronic Payment Protocols

Yi LIU*, Nonmember, Qingkun MENG†, Member, Xingtong LIU*,a, Jian WANG†, Lei ZHANG†, and Chaojing TANG†, Nonmembers

SUMMARY Electronic payment protocols provide secure service for electronic commerce transactions and protect private information from malicious entities in a network. Formal methods have been introduced to verify the security of electronic payment protocols; however, these methods concentrate on the accountability and fairness of the protocols, without considering the impact caused by timeliness. To make up for this deficiency, we present a formal method to analyze the security properties of electronic payment protocols, namely, accountability, fairness and timeliness. We add a concise time expression to an existing logical reasoning method to represent the event time and extend the time characteristics of the logical inference rules. Then, the Netbill protocol is analyzed with our formal method, and we find that the fairness of the protocol is not satisfied due to the timeliness problem. The results illustrate that our formal method can analyze the key properties of electronic payment protocols. Furthermore, it can be used to verify the time properties of other security protocols.

key words: electronic payment protocol, formal analysis, accountability, fairness, timeliness, logical reasoning

1. Introduction

In recent years, the explosion of services provided over the Internet has had a substantial impact on daily life. The transfer of private and financial information over networks is a great challenge. Chen proposed a proof methodology to verify secure routing protocols [1]. However, another category of network protocols to protect legitimate interests between traders also needs to be verified with additional security properties. Electronic payment protocols provide technical assurance for secure electronic commerce. Sensitive information, such as credit card numbers and passwords, depends on the security of electronic payment protocols, which work as transport channels. Research on the security of electronic payment protocols has received much attention in academic and industrial areas [2].

Compared with other security protocols, accountability, fairness and timeliness are additional security properties of electronic payment protocols. Accountability can provide sufficient evidence to resolve possible future disputes after the execution of the protocol [3]. Accountability means that all parties cannot repudiate what they have done. Fairness means that neither of the participants has a chance to obtain advantages over the other by misbehaving, which means that either both participants receive what they expect or nothing. Timeliness provides an interval constraint during each step in the protocol to avoid time differences that can be utilized by attackers.

Formal analysis is an effective method to verify electronic payment protocols due to its strict and effective characteristics. However, the current formal methods for the analysis of electronic payment protocols lack descriptions and analyses of timeliness. Our approach focuses on the description and analysis of the three security properties mentioned above. We enhance the ability of an existing logical reasoning method by adding a concise time expression. The logical reasoning part of the objective proof is based on the Qin-Zhou logic method [4], [5], and the time calculus component utilizes algebraic methods and set theory. The logical and algebraic methods are independent: they do not interfere with each other or undermine the correctness of the original method [6]. The Netbill protocol is analyzed with the our method, and the result show that the protocol does not satisfy fairness because of timeliness defects. Then, we show that the defect can be fixed with careful specification of the event time and waiting time.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes the concepts and definitions of our logical method. The logical analysis procedure is introduced in Sect. 4. The analysis process of the Netbill protocol is illustrated in Sect. 5. Section 6 concludes the paper and outlines our future studies.

2. Related Work

Formal methods have been used for the security analysis of electronic payment protocols for decades [7]. They can be divided into three categories: logical reasoning, model checking and theorem proving.

2.1 Logical Reasoning

Logical reasoning is the origin of formal methods for analyzing electronic payment protocols. Kailar logic [7] was the first analysis method designed for electronic payment protocols and was mainly used to analyse accountability. However, it ignored fairness in electronic payment protocols. Volker extended Autlog logic to analyze accountability [8]. The famous Payword and SET protocols were analyzed as examples. Qing-Zhou logic was proposed for the
analysis of accountability and fairness [4], [5]. Li added a time factor to SVO logic to enable it to analyze the timeliness of protocols [9]. Wen proposed a modeling and analysis method for electronic payment protocols based on game logic [10]. Chen combined logical reasoning with a strand space model and introduced a new logical analysis method for electronic payment protocols [11]. A method applying Kailar logic in compositional analysis was presented by Gao to analyze the accountability and fairness of electronic payment protocols [12].

2.2 Model Checking

The characteristics of model checking are easy to manipulate. Kremer applied the model-checker MOCHA, which supports alternating transition systems and alternating temporal logic, to analyze accountability [13]. Xie utilized finite automata to analyze the ISI and IBS protocols [14]. Guo combined a communication finite state machine with new logic rules based on Qing-Zhou logic to analyze the security properties of electronic payment protocols [15]. Liu proposed an extended deterministic finite automaton that can analyze security properties, such as accountability and fairness [16]. Dreier modeled e-cash systems in the applied π-calculus and used ProVerif as the verification tool [17]. Nevertheless, because the state space of the model checking method was limited, even if no attack method is found, the correctness of the protocol cannot be verified.

2.3 Theorem Proving

Theorem proving is regarded as an accurate method for cryptographic protocol security analysis. Papa integrated logic with process calculus to analyze electronic payment protocols [18]. Ouyang used colored Petri nets to analyze the Internet open trading protocol [19]. Bella analyzed the purchase protocol of SET with Isabelle and the inductive method [20]. Gutman applied the strand space method to analyze the fairness of fair-exchange protocols [21]. Guo proposed a technique to model and verify fair-change electronic payment protocols [22]. However, the theorem proving method is complicated, and it is difficult to verify complex protocols.

The above methods analyze electronic payment protocols without consideration of timeliness, which is a crucial security property. Since most researchers concentrated on accountability and fairness of electronic payment protocols in the past, they didn’t realize that timeliness of electronic payment protocols also has an impact on the security of protocols. Some researchers has added a time factor to SVO logic to analyze general security protocols, but they didn’t use it to analyze electronic payment protocols. To the best of our knowledge, the formal method presented in our work is the first attempt to introduce timeliness to the security analysis of electronic payment protocols.

3. Model and Specifications

The definitions and symbols used in the formal method are denoted as follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B</td>
<td>Parties participating in the protocol.</td>
</tr>
<tr>
<td>TTP</td>
<td>Trusted third party.</td>
</tr>
<tr>
<td>m</td>
<td>Message transferred in protocol.</td>
</tr>
<tr>
<td>n</td>
<td>Message m is concatenated with message n.</td>
</tr>
<tr>
<td>m_k</td>
<td>Ciphertext of message m encrypted with a secret key K.</td>
</tr>
<tr>
<td>m_k</td>
<td>A digital signature of message m signed with A’s private key K_A^{-1}.</td>
</tr>
<tr>
<td>h(m)</td>
<td>Hash value of message m.</td>
</tr>
<tr>
<td>K_AB</td>
<td>A shared session key between participants A and B.</td>
</tr>
<tr>
<td>S_i.A</td>
<td>A signature on the i\textsuperscript{th} transferred message in protocol by the sender A.</td>
</tr>
<tr>
<td>EOO</td>
<td>The non-repudiation evidence that is provided to the receiver in electronic payment protocols, which is used to prove that the sender has sent the message.</td>
</tr>
<tr>
<td>EOR</td>
<td>The non-repudiation evidence that is provided to the sender in electronic payment protocols, which is used to prove that the receiver has received the message.</td>
</tr>
<tr>
<td>T</td>
<td>Time of event.</td>
</tr>
</tbody>
</table>

3.1 Time System

We add a condition after the logical expression to define the time when events occur, for example, A \rightarrow m at T. T is a time expression [23]. This definition describes when parties send or receive messages. The time expression is defined as follows:

1. \( x \) stands for a constant time element.
2. \( X \) stands for a variable time element.
3. \( X[TS] \) is a time binding expression, where \( TS \) is the scope of \( X \).
4. \( [T] \) is a time expression, where \( T \) is a time-binding expression.

The constant time element is represented by \( t \), and the variable time element is represented by \( T \). A time-binding expression is represented by a variable time element \( X \) with a specific constant time element \( t (t \in TS) \). In logic formulas, the time expression \( [X[I] \) can be abbreviated as \( [X] \), and \( [X][x] \) can be abbreviated as \( [x] \) if \( x \) is a constant time element or a variable time element with bound value. The value of a variable time element is bound to the first operation in its formula.

3.2 Protocol and Environment

\( TTP \) (trusted third party) stands for a party that can be fully trusted. A bank or arbitration organization can act as a \( TTP \). Usually, all parties are considered to be dishonest except for
the TTP, which may interrupt the execution of the protocols arbitrarily.

Whether a communication channel is reliable depends on the environment in which it operates. In general, the communication channel between general parties is considered to be unreliable, whereas the channel between the TTP and other parties is recoverable, which means the message will be transferred eventually.

A protocol statement describes what message should be sent or received in the current round:

\[ A \rightarrow B : m \text{ at } T \] means \( A \) sent a message \( m \) to \( B \) at \( T \).

3.3 Possession Set

\( O_a \) represents the possession set of party \( A \) in the protocol. Assuming the protocol begins at \( T_0 \), the initial possession set of \( A \) is \( O_a(T_0) \). \( O_a(T_x) \) represents the possession set of \( A \) at \( T_x \), and \( O_a(T_x) \) stands for the final possession set of \( A \) upon completion of the protocol. The possession set of \( A \) contains the information inherited from the last step and the message that is received or generated at present. The possession set varies consecutively with the execution of the protocol until \( O_a = O_a(T_e) \).

The possession set of \( A \) changes from \( O_a(T_y) \) to \( O_a(T_x) \cap (T_y < T_x) \), where \( T_y \) indicates the moment before \( T_x \). It varies as follows:

1. There are two possible results of \( O_a(T_x) \) when the execution of protocol statement is \( A \rightarrow B : m \) at \( T_x \). If \( m \notin O_a(T_y) \), which means \( m \) is a new message generated by \( A \), \( O_a(T_x) = O_a(T_y) \cup \{m\} \). Otherwise, \( O_a(T_x) = O_a(T_y) \) when \( m \in O_a(T_y) \).
2. When the execution of the protocol statement is \( B \rightarrow A : m \) at \( T_x \), and \( m \notin O_a(T_y), O_a(T_x) = O_a(T_y) \cup \{m\} \). Otherwise, \( O_a(T_x) = O_a(T_y) \).

4. Logical Analysis Method

4.1 Logic Component

Our method comprises five logic components:

1. (1) A \( > x \). A can prove formula \( x \) is satisfied without leaking any secret.
2. (2) \( A \rightarrow m \) at \( T \). \( A \) sends a message \( m \) at \( T \) to his recipient through their communication channel regulated by protocols. The following implication is also established as usual.

\[ A \rightarrow (m, n) \text{ at } T \Rightarrow A \rightarrow m \text{ at } T \]

We can infer that \( A \) sends message \( m \) at \( T \) based on the fact that \( A \) sends message \( (m, n) \) at \( T \).

3. (3) A \( \equiv m \). A possesses message \( m \).

4. (4) A \( \leftarrow m \) at \( T \). A receives message \( m \) at \( T \). The following implication is established as the second component:

\[ A \leftarrow (m, n) \text{ at } T \Rightarrow A \leftarrow m \text{ at } T \]

5. (5) \( K_A \). \( K_A \) is the public key of \( A \), which is used to verify the message signed by its private key \( K_A^{-1} \).

6. (6) \( A \leftrightarrow B. K_{AB} \) is the shared session key between \( A \) and \( B \).

4.2 Axiom System

The axiom system includes 1 inference rule and 8 axioms. The inference rule is depicted as:

\[ (\vdash \varphi \land (\vdash (\varphi \Rightarrow \psi)) \Rightarrow \vdash \psi \]

\( \vdash \psi \) can be obtained from \( \vdash \varphi \) and \( \vdash (\varphi \Rightarrow \psi) \). \( \Gamma \vdash \varphi \) indicates that \( \varphi \) can be deduced from the formula set \( \Gamma \). \( \vdash \varphi \) indicates that \( \varphi \) is a theorem, which means \( \varphi \) is established all the time. The inference rule above indicates that \( \psi \) is a theorem if \( \varphi \) is a theorem, and \( \varphi \) implies \( \psi \). The 8 axioms are as follows:

A1. A \( \ni (x \land y) \Rightarrow A \ni x \land y \)

A2. A \( \ni (x \land y) \Rightarrow A > y \)

A3. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( [T_y,T_x] \ni T_y \leq T_x \)

A4. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( [T_y,T_x] \ni T_y \leq T_x \)

A5. A \( \ni K_{BA} \) at \( T_x \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A6. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A7. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A8. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A9. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A10. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A11. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A12. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A13. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A14. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A15. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A16. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A17. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A18. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A19. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)

A20. A \( \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni \{m\}_{K_{BA}} \ni A > B \ni m \) at \( max(T_x,T_y) \)
message \( m \) at \( T \).

The proof of the protocol properties is divided into two parts: logical reasoning and time calculus. The objective is to verify whether the result obtained from logical reasoning satisfies the time constraints specified by the time calculus.

4.3 Protocol Analysis Procedure

The protocol analysis consists of the 5 steps shown in Fig. 1:

5. Netbill Protocol Analysis

The Netbill protocol was proposed by Professor J.D. Tygar from Carnegie Mellon University for digital goods transactions. The protocol consists of three participants: customer, merchant, and Netbill server [24]. Netbill protocol works on the application layer which is based on the establishment of shared session keys. For concentrating on Netbill protocol and simplifying analysis process, we assume that shared session keys among \( C \), \( M \) and \( N \) have been established through a key exchange protocol such as the Diffie-Hellman protocol [25]. The main steps are as follows:

1. \( C \rightarrow M : \{PRD, TID, \text{Sig}_C\}_{K_{CM}} \) at \( T_1 \)
2. \( M \rightarrow C : \{ProductID, Price, TID, \text{Sig}_M\}_{K_{CM}} \) at \( T_2 \)
3. \( C \rightarrow M : \{TID, \text{Sig}_C\}_{K_{CM}} \) at \( T_3 \)
4. \( M \rightarrow C : \{\text{Goods}_k, h(\text{Goods}_k), \text{EPOID}, \text{Sig}_M\}_{K_{CM}} \) at \( T_4 \)
5. \( C \rightarrow M : \{\text{EOO}\}_{K^{-1}_C}, \text{Sig}_C \) at \( T_5 \)
6. \( M \rightarrow N : \{\text{Result}, C, M, \text{EOPOD}, k\}_{K_{CM}} \) at \( T_6 \)
7. \( N \rightarrow M : \{\text{EOO}\}_{K^{-1}_N} \)
8. \( M \rightarrow C \):
   - \( \{\text{EOO}\}_{K^{-1}_C} \)
   - \( \{\text{EOPOD}, \text{MAC}, k\}_{K_{CM}}, \text{Sig}_{GN}\}_{K_{MN}} \) at \( T_7 \)
9. \( M \rightarrow C \):
   - \( \{\text{EOO}\}_{K^{-1}_C} \)
   - \( \{\text{EOPOD}, \text{MAC}, k\}_{K_{CM}}, \text{Sig}_{BN}\}_{K_{MN}} \) at \( T_8 \)

For instance, \( \text{Sig}_C = \{h((\text{PRD}, \text{TID}))\}_{K^{-1}_C} \) and \( \text{Sig}_M = \{h((\text{ProductID}, \text{Price}, \text{TID}))\}_{K^{-1}_C} \). PRD is the product request data. \( \text{TID} \) is the transaction identification, \( \text{ProductID} \) is the product identification and \( \text{Price} \) stands for the price of the commodity. \( \text{Goods} \) is the specific content of transmitted goods. \( k \) represents the secret key used to encrypt and decrypt the transmitted goods. \( \text{EPOID} \) is a unique identifier for the transaction in the Netbill server database. \( \text{EPO} = \{\text{EPOID}, \text{ProductID}, \text{Price}, C, M, h(\text{Goods}_k)\} \) represents an electronic purchase order. \( \text{MAC} \) and \( \text{MAcc} \) stand for the customer and merchant accounts respectively. The encryption component includes a payment instruction that can only be read by the Netbill server, such as the customer account. \( \text{Receipt} = \{\text{Result}, C, M, \text{EPOID}, k\} \) stands for the receipt returned from the Netbill server, where \( \text{Result} \) indicates whether to accept the payment. The Netbill protocol analysis procedure is detailed in the following paper.

5.1 The Initial Possession Sets

At the beginning of the protocol, the initial states of \( C \) and \( M \) are

\[
O_C(T_0) = \{K^{-1}_C, K_C, K_M, K_N, K_{CM}, K_{CN}\}
\]

\[
O_M(T_0) = \{K^{-1}_M, K_M, K_C, K_{CN}, K_{CM}, K_{MN}\}
\]

\[
C > (\overset{K_M}{M} \xrightarrow{K_N} N, C \overset{K_{CM}}{\leftrightarrow} M, C \overset{K_{CM}}{\leftrightarrow} N)
\]

\[
M > (\overset{K_C}{C} \xrightarrow{K_N} N, C \overset{K_{CM}}{\leftrightarrow} M, M \overset{K_{MN}}{\leftrightarrow} N)
\]

5.2 The Credible Assumptions

T1: \( A > N \rightarrow k \Rightarrow A > B \rightarrow k \)

The Netbill server \( N \) is assumed to be a fully trusted third party that obeys the protocol specification strictly. \( N \) will do as the 7th step in Netbill protocol to send \( k \) if and only if he receives \( k \) from \( B \). \( N \) will never send any messages to deviate from the protocol. So if \( A \) can prove that \( N \) has sent \( k \), he can prove that the other party \( B \) has sent \( k \).

T2: \( A > B \rightarrow h(m) \Rightarrow A > B \rightarrow m \)

According to the Netbill protocol, \( h(m) \) is transmitted for the checksum of message \( m \). Only the owner of massage \( m \) has the ability to calculate its checksum. The sender can calculate the checksum only if he has owned message \( m \). Then, if \( A \) can prove that \( B \) has sent \( h(m) \), \( A \) can prove that \( B \) has sent message \( m \).

5.3 EOO and EOR

In Netbill protocol, \( \text{EOO} \) is a message set to prove that \( M \) has sent the product \( \text{Goods} \). \( C > M \rightarrow \text{Goods} \) can be deduced from \( \text{EOO} \in O_C(T_r) \). \( \text{EOR} \) is a message set to prove that \( C \) has received the product \( \text{Goods} \). \( M > C \rightarrow \text{Goods} \) can be deduced from \( \text{EOR} \in O_M(T_r) \). We choose \( \text{EOO} \) and \( \text{EOR} \) as below and check whether they satisfy the requirement of accountability.
\[ EOO = \langle \{ h(\text{Goods}) \}\rangle_{K_{CM}}, \{ k \}_{K_{CM}} \]  
\[ EOR = \langle \{ h(\text{Goods}) \}\rangle_{K_{CM}^{-1}}, \{ k \}_{K_{CM}^{-1}} \]  

Assume that \( EOO \in O_C(T_e) \) is established at the end of the protocol. Then, \( \langle h(\text{Goods}) \rangle_{K_{CM}}, \{ k \}_{K_{CM}} \in O_C(T_e) \) is satisfied, which means \( C \ni \langle h(\text{Goods}) \rangle_{K_{CM}} \) at \( T_e \) and \( C \ni \{ k \}_{K_{CM}} \) at \( T_e \).

Since \( C \ni \{ h(\text{Goods}) \\}_{K_{CM}} \) at \( T_e \), \( C > C \xrightarrow{K_{CM}} M \) and axiom A4, then \( C > M \rightarrow h(\text{Goods}) \) at \( [T_a]T_a \leq T_e \). According to T2, we obtain

\[
C > M \rightarrow \{ \text{Goods} \} \text{ at } [T_a]T_a \leq T_e
\]  

(1)

Since \( C > \{ k \}_{K_{CM}} \) at \( T_e \), \( C \xrightarrow{K_{CM}} N \) and axiom A3; therefore, \( C > N \rightarrow k \) at \( [T_b]T_b \leq T_e \). According to the credible assumption T1, we obtain

\[
C > M \rightarrow k \text{ at } [T_b]T_b \leq T_e.
\]  

(2)

From formulas (1) and (2) and axiom A5, we obtain

\[
C > M \rightarrow \text{Goods} \text{ at } max(T_a, T_b) \cap [T_a]T_a \leq T_e \cap [T_b]T_b \leq T_e.
\]  

(3)

Assume that \( EOR \in O_M(T_e) \) is established when the protocol finishes, which means \( M \ni \{ h(\text{Goods}) \}\rangle_{K_{CM}^{-1}} \) at \( T_e \) and \( M \ni \{ k \}_{K_{CM}^{-1}} \) at \( T_e \) are satisfied. Then, according to \( M \xrightarrow{K_{CM}^{-1}} N \), axiom A3, A8 and credible assumption T1, we obtain

\[
M > C \ni k \text{ at } [T_y]T_y \leq T_e.
\]  

(4)

Since \( M > K_{CM} \xrightarrow{C} \), according to axiom A3, A8 and credible assumption T2, we get \( M > C \ni \{ \text{Goods} \} \) at \( [T_a]T_a \leq T_e \). Due to formula (4) and axiom A6, we obtain

\[
M > C \ni \text{Goods} \text{ at } max(T_y, T_b) \cap [T_y]T_y \leq T_e \cap [T_b]T_b \leq T_e.
\]  

(5)

Hence, the choices of EOO and EOR in the Netbill protocol satisfy the requirement of accountability.

5.4 Analysis of Accountability

Verify whether \( C \) and \( M \) can obtain the appropriate evidence at the end of protocol. EOO is not sent to \( C \) as a whole. \( \{ h(\text{Goods}) \}\rangle_{K_{CM}} \) is sent to \( C \) during the 4th step of the Netbill protocol as the first part of EOO. After the fourth step of the protocol, \( O_C(T_4) = O_C(T_3) \cup \{ h(\text{Goods}) \}\rangle_{K_{CM}} \cap [T_4]T_4 \leq T_e \), and \( \{ h(\text{Goods}) \}\rangle_{K_{CM}} \in O_C(T_4) \).

Since \( k \) is included in Receipt, it is signed by \( N \) in the 7th step and sent to \( C \) during the last step of the Netbill protocol. Then \( C \) could decrypt the last message \( \{ \{ \text{Receipt} \}}_{K_{CM}} \), \{ EPOID, C\text{Accr}, k \}_{K_{CM}}, S_{\text{iq}} \text{ with his shared key K}_{CM} \) and obtain \( \{ k \}_{K_{CM}} \). When the last step of the protocol is completed, \( O_C(T_8) = O_C(T_7) \cup \{ \{ \text{Receipt} \}_k \}_{K_{CM}} \cap [T_8]T_8 \leq T_e \). Because \( C \ni K_{CM} \), we obtain \( \{ \text{Receipt} \}_k \in O_C(T_7) \), and \( \{ k \}_{K_{CM}} \in O_C(T_8) \).

6. Conclusion

The Netbill protocol analysis results show that the protocol does not satisfy fairness because of timeliness defects. The
analysis procedure illustrates how our method can be applied to analyze the temporal relation among events in electronic payment protocols. It is an integrated approach rather than a simple logic method. The formal method proposed in this paper can guide the design of electronic payment protocols and fix the defects of the original protocols.

The next step of our research is to analyze additional electronic payment protocols that are widely used in electronic commerce with our method. Furthermore, we will study automated analysis tools that make it convenient to design and analyze electronic payment protocols.

References

Jian Wang received his Ph.D. degree from National University of Defense Technology, China, in 2008. He is currently a professor in National University of Defense Technology. His main research interests are wireless network security, quantum communication and computer network.

Lei Zhang received Ph.D. degree in information and communication engineering from National University of Defense technology (NUDT) in 2010. Currently, he is an associate professor of communication engineering at NUDT. His main research interests are interplanetary networks and space communication security, cyberspace security, industrial control systems security and Internet of things.

Chaojing Tang received his Ph.D. degree from National University of Defense Technology, China, in 2003. He is currently a professor in National University of Defense Technology. His main research includes information security, electromagnetic countermeasure and software vulnerabilities.