PAPER

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

Manuscript received March 20, 2018.
Manuscript revised May 14, 2018.
Manuscript publicized June 19, 2018.

[†]The authors are with College of Electronic Science, National University of Defense Technology, Changsha, 410073 China.
a) E-mail: liuxingtong@nudt.edu.cn

DOI: 10.1587/transinf.2018EDP7108

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 automaton 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]. Guttman 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:
A,B – Parties participating in the protocol.
TTP – Trusted third party.
m – Message transferred in protocol.
(m, n) – Message m is concatenated with message n.
$\{m\}_K$ – Ciphertext of message m encrypted with a secret key K.
K_A – The public key of party A, which is used to verify the digital signature of A.
K_A^{-1} – The private key corresponding to K_A.
$\{m\}_{K_A^{-1}}$ – A digital signature of message m signed with A's private key K_A^{-1}.
$h(m)$ – Hash value of message m.
K_{AB} – A shared session key between participants A and B.
Sig_{iA} – A signature on the ith transferred message in protocol by the sender A.
EOO – 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.
EOR – 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.
T – Time of event.

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:

$A \rightarrow B : m$ 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 $O_a(T_e)$ 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) $A \succ x$. A can prove formula x is satisfied without leaking any secret.

(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) $A \ni m$. A possesses message m.

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

(6) $A \xleftrightarrow{K_{AB}} B$. 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:

$$(\vdash \varphi) \cap (\vdash (\varphi \Rightarrow \psi)) \Rightarrow \vdash \psi.$$

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

A1. $A \succ x \cap A \succ y \Rightarrow A \succ (x \wedge y)$

If A can prove formula x and formula y simultaneously, A can prove the intersection of x and y.

A2. $A \succ x \cap (x \Rightarrow y) \Rightarrow A \succ y$

If A can prove formula x and x implies y, then A can prove formula y.

A3. $A \ni \{m\}_{K_B^{-1}}$ at $T_x \cap A \succ \xrightarrow{K_B} B$ at $T_x \Rightarrow A \succ B \rightarrow m$ at $[T_y | T_y \leq T_x]$

If A possesses a ciphertext $\{m\}_{K_B^{-1}}$ signed with the private key of B at T_x and A can prove K_B is the public key of B, then A can prove B sent the message m at a moment T_y which is before T_x. Because K_B^{-1} is only known to B and no one else has the ability to forge a signature of B.

A4. $A \leftarrow \{m\}_{K_{AB}}$ at $T_x \cap A \succ A \xleftrightarrow{K_{AB}} B$ at $T_x \Rightarrow A \succ B \rightarrow m$ at $[T_y | T_y \leq T_x]$

If A receives a ciphertext $\{m\}_{K_{AB}}$ at T_x and A can prove K_{AB} is the shared session key between A and B, then A can prove B sent the message m at a moment T_y which is before T_x. Because no one has the ability to encrypt messages with K_{AB} except for A and B.

A5. $A \succ B \rightarrow \{m\}_K$ at $T_x \cap A \succ B \rightarrow K$ at $T_y \Rightarrow A \succ B \rightarrow m$ at $\max(T_x, T_y)$

If A can prove B sent a ciphertext $\{m\}_K$ encrypted with a secret key K at T_x and A can prove B sent the secret key K at T_y, then A can prove B sent the message m at $\max(T_x, T_y)$. Because anyone in the network can decrypt the ciphertext $\{m\}_K$ with the secret key K, It works as B sending the message m directly.

A6. $A \leftarrow \{m\}_K$ at $T_x \cap A \ni K$ at $T_y \Rightarrow A \leftarrow m$ at $\max(T_x, T_y)$

If A receives a ciphertext $\{m\}_K$ encrypted with the secret key K at T_x and A possesses K at T_y, then A receives the message m at $\max(T_x, T_y)$. Since A can utilize the secret key K to decrypt the ciphertext $\{m\}_K$ when he gets these two messages.

A7. $A \leftarrow m$ at $T \Rightarrow A \ni m$ at T

A possesses message m at T if A receives the message m at T.

A8. $A \rightarrow m$ at $T \Rightarrow A \ni m$ at T

Only if A possesses message m at T, he can send the

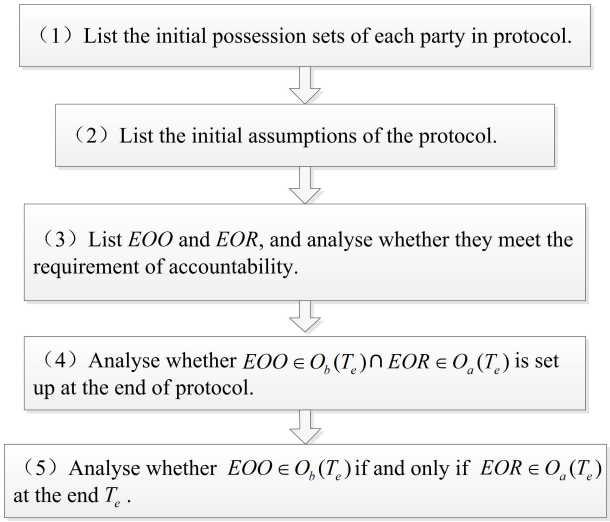

Fig. 1 Protocol analysis procedure

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) $C \rightarrow M : \{PRD, TID, Sig_{1C}\}_{K_{CM}}$ at T_1

(2) $M \rightarrow C : \{ProductID, Price, TID, Sig_{2M}\}_{K_{CM}}$ at T_2

(3) $C \rightarrow M : \{TID, Sig_{3C}\}_{K_{CM}}$ at T_3

(4) $M \rightarrow C : \{\{Goods\}_k, h(\{Goods\}_k), EPOID, Sig_{4M}\}_{K_{CM}}$ at T_4

(5) $C \rightarrow M : \{\{EPO\}_{K_C^{-1}}, Sig_{5C}\}_{K_{CM}}$ at T_5

(6) $M \rightarrow N : \{\{\{EPO\}_{K_C^{-1}}, MAcct, k\}_{K_M^{-1}}, Sig_{6M}\}_{K_{MN}}$ at T_6

(7) $N \rightarrow M :$
$\{\{Receipt\}_{K_N^{-1}}, \{EPOID, CAcct, k\}_{K_{CN}}, Sig_{7N}\}_{K_{MN}}$ at T_7

(8) $M \rightarrow C :$
$\{\{Receipt\}_{K_N^{-1}}, \{EPOID, CAcct, k\}_{K_{CN}}, Sig_{8M}\}_{K_{CM}}$ at T_8

C, M and N represent the customer, the merchant and the Netbill server, respectively. Sig works as a signature on the message for authentication and integrity in each step. For instance, $Sig_{1C} = \{h((PRD, TID))\}_{K_C^{-1}}$ and $Sig_{2M} = \{h((ProductID, Price, TID))\}_{K_M^{-1}}$. PRD is the product request data. TID is the transaction identification, $ProductID$ is the product identification and $Price$ stands for the price of the commodity. $Goods$ is the specific content of transmitted goods. k represents the secret key used to encrypt and decrypt the transmitted goods. $EPOID$ is a unique identifier for the transaction in the Netbill server database. $EPO = \{EPOID, ProductID, Price, C, M, h(\{Goods\}_k)\}$ represents an electronic purchase order. $CAcct$ and $MAcct$ 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. $Receipt = \{Result, C, M, EPOID, k\}$ stands for the receipt returned from the Netbill server, where $Result$ 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

$$O_C(T_0) = \{K_C^{-1}, K_C, K_M, K_N, K_{CM}, K_{CN}\}$$

$$O_M(T_0) = \{K_M^{-1}, K_M, K_C, K_N, K_{CM}, K_{MN}\}$$

$$C \succ (\xrightarrow{K_M} M, \xrightarrow{K_N} N, C \xleftrightarrow{K_{CM}} M, C \xleftrightarrow{K_{CN}} N)$$

$$M \succ (\xrightarrow{K_C} C, \xrightarrow{K_N} N, C \xleftrightarrow{K_{CM}} M, M \xleftrightarrow{K_{MN}} N)$$

5.2 The Credible Assumptions

T1:$A \succ N \rightarrow k \Rightarrow A \succ B \rightarrow k$

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:$A \succ B \rightarrow h(m) \Rightarrow A \succ B \rightarrow m$

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,EOO is a message set to prove that M has send the product $Goods$. $C \succ M \rightarrow Goods$ can be deduced from $EOO \in O_C(T_e)$. EOR is a message set to prove that C has received the product $Goods$. $M \succ C \ni Goods$ can be deduced from $EOR \in O_M(T_e)$. We choose EOO and EOR as below and check whether they satisfy the requirement of accountability.

EOO=$(\{h(\{Goods\}_k)\}_{K_{CM}}, \{k\}_{K_N^{-1}})$

EOR=$(\{h(\{Goods\}_k)\}_{K_C^{-1}}, \{k\}_{K_N^{-1}})$

Assume that $EOO \in O_C(T_e)$ is established at the end of the protocol. Then, $(\{h(\{Goods\}_k)\}_{K_{CM}}, \{k\}_{K_N^{-1}}) \in O_C(T_e)$ is satisfied, which means $C \ni \{h(\{Goods\}_k)\}_{K_{CM}}$ at T_e and $C \ni \{k\}_{K_N^{-1}}$ at T_e.

Since $C \ni \{h(\{Goods\}_k)\}_{K_{CM}}$ at T_e, $C \succ C \overset{K_{CM}}{\longleftrightarrow} M$ and axiom A4, then $C \succ M \rightarrow h(\{Goods\}_k)$ at $[T_\alpha | T_\alpha \leq T_e]$. According to T2, we obtain

$$C \succ M \rightarrow \{Goods\}_k \quad \text{at} \quad [T_\alpha | T_\alpha \leq T_e] \tag{1}$$

Since $C \ni \{k\}_{K_N^{-1}}$ at T_e, $C \succ \overset{K_N}{\longrightarrow} N$ and axiom A3; therefore, $C \succ N \rightarrow k$ at $[T_\beta | T_\beta \leq T_e]$. According to the credible assumption T1, we obtain

$$C \succ M \rightarrow k \quad \text{at} \quad [T_\beta | T_\beta \leq T_e]. \tag{2}$$

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

$$C \succ M \rightarrow Goods \quad \text{at} \quad max(T_\alpha, T_\beta) \cap [T_\alpha | T_\alpha \leq T_e] \cap [T_\beta | T_\beta \leq T_e]. \tag{3}$$

Assume that $EOR \in O_M(T_e)$ is established when the protocol finishes, which means $M \ni \{h(\{Goods\}_k)\}_{K_C^{-1}}$ at T_e and $M \ni \{k\}_{K_N^{-1}}$ at T_e are satisfied. Then, according to $M \succ \overset{K_N}{\longrightarrow} N$, axiom A3,A8 and credible assumption T1, we obtain

$$M \succ C \ni k \quad \text{at} \quad [T_\gamma | T_\gamma \leq T_e]. \tag{4}$$

Since $M \succ \overset{K_C}{\longrightarrow} C$, according to axiom A3,A8 and credible assumption T2, we get $M \succ C \ni \{Goods\}_k$ at $[T_\theta | T_\theta \leq T_e]$. Due to formula (4) and axiom A6, we obtain

$$M \succ C \ni Goods \quad \text{at} \quad max(T_\gamma, T_\theta) \cap [T_\gamma | T_\gamma \leq T_e] \cap [T_\theta | T_\theta \leq T_e]. \tag{5}$$

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. $\{h(\{Goods\}_k)\}_{K_{CM}}$ 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, $O_C(T_4) = O_C(T_3) \cup \{h(\{Goods\}_k)\}_{K_{CM}} \cap [T_4 | T_4 \leq T_e]$, and $\{h(\{Goods\}_k)\}_{K_{CM}} \in O_C(T_e)$.

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 $\{\{Receipt\}_{K_N^{-1}}, \{EPOID, CAcct, k\}_{K_{CN}}, Sig_{8M}\}_{K_{CM}}$ with his shared key K_{CM} and obtain $\{k\}_{K_N^{-1}}$. When the last step of the protocol is completed, $O_C(T_8) = O_C(T_7) \cup \{\{Receipt\}_{K_N^{-1}}\}_{K_{CM}} \cap [T_8 | T_8 \leq T_e]$. Because $C \ni K_{CM}$, we obtain $\{Receipt\}_{K_N^{-1}} \in O_C(T_e)$, and $\{k\}_{K_N^{-1}} \in O_C(T_e)$. $\{h(\{Goods\}_k)\}_{K_{CM}}$ and $\{k\}_{K_N^{-1}}$ are combined to generate EOO by C. Then, $EOO \in O_C(T_e)$ is satisfied.

Similarly, according to the fifth step of the protocol, we obtain $\{h(\{Goods\}_k)\}_{K_C^{-1}} \ni O_M(T_e)$. $\{k\}_{K_N^{-1}} \in O_M(T_e)$ is obtained after the seventh step. Then, we get $EOR \in O_M(T_e)$.

Therefore, $EOO \in O_C(T_e) \cap EOR \in O_M(T_e)$ is satisfied when the protocol finishes.

5.5 Analysis of Fairness

The fairness objective is:

$$EOO \in O_C(T_k) \quad \text{if and only if} \quad EOR \in O_M(T_k).$$

Everyone in the protocol waits for the next step after the previous step in completed. If no response is received, the protocol terminates after a certain period t and clears the previous protocol records. To ensure fairness, the following conditions have to be satisfied:

$$\begin{aligned} M \rightarrow \{\{k\}_{K_M^{-1}}\}_{K_{MN}} \quad &\text{at} \quad T_x \quad \cap \\ M \leftarrow \{k\}_{K_N^{-1}} \quad &\text{at} \quad T_y \quad \cap \\ (T_x \leq T_y \leq T_x + t_M) \end{aligned} \tag{6}$$

$$\begin{aligned} C \rightarrow \{h(\{Goods\}_k)\}_{K_C^{-1}} \quad &\text{at} \quad T_x \quad \cap \\ C \leftarrow \{k\}_{K_N^{-1}} \quad &\text{at} \quad T_y \quad \cap \\ (T_x \leq T_y \leq T_x + t_C) \quad . \end{aligned} \tag{7}$$

t_M is the waiting time after M executes the 6th step of the protocol. t_C is the waiting time after C executes the 5th step. Since N is in full accordance with the regulation of the protocol, formula (6) must be established.

In formula (7), T_x=T_5, and T_y=$T_7 = T_5 + t_5 + t_6$, where t_5 and t_6 are the time delays after the 5th and 6th steps. Therefore, we must ensure $t_5 + t_6 \leq t_C$ to establish formula (7). However, there are no restrictions on the relationship among t_5, t_6 and t_C. It is possible to make $t_5 + t_6 > t_C$, regardless of what constant t_C is specified. For example, if C performs the 5th step in accordance with the regulation, M could send $\{\{EPO\}_{K_C^{-1}}\}_{K_{MN}}$ to N after its timeout to acquire the evidence to prove that C has received the product $Goods$. However, C has deleted $\{Goods\}_k$ because of the timeout. Although C has received $Receipt$ and the key k, he could not decrypt the ciphertext to obtain the product $Goods$.

Formula (7) can not be satisfied, which means the protocol does not achieve the fairness objective. This problem occurs because there are no specific constraints on the relevant event time in the protocol specification. To make up for this defect, the event time and waiting time should be carefully regualted in the protocol specification.

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 mehtod 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

[1] C. Chen, L. Jia, H. Xu, C. Luo, W. Zhou, and B.T. Loo, "A Program Logic for Verifying Secure Routing Protocols," Logical Methods in Computer Science, vol.11, no.4, 2015.
[2] P. McCorry, S.F. Shahandashti, and F. Hao, "Refund attacks on Bitcoin's Payment Protocol," Barbados: 20th Financial Cryptography and Data Security, vol.9603, pp.581–599, 2016.
[3] R. Küsters, T. Truderung, and A. Vogt, "Accountabiliy:Definition and Relationship to Verifiability," The 17th ACM Conference on Computer and Communications Security, pp.526–535, 2010.
[4] S. Qing, "A Formal Method for Analyzing Electronic Commerce Protocols," J. Software, 2005.
[5] D. Zhou, S. Qing, and Z. Zhou, "A new approach for the analysis of electronic payment protocols," J. Software, 2001.
[6] Y. Liu, X. Liu, J. Ye, and C. Tang, "Formal Analysis of Timeliness in Electronic Commerce Protocols," 2016 Progress in Electromagnetic Research Symposium (PIERS), 2016.
[7] R. Kailar, "Accountability in Electronic Commerce Protocols," Piscataway, NJ, USA: IEEE Trans. Softw. Eng., vol.22, no.5, pp.313–328, 1996
[8] K. Volker and N. Heike, "A sound logic for analysing electronic payment protocols," The 5th European Symposium on Research in Computer Security, 1998.
[9] B. Li and J. Luo, "Formal Analysis of Timeliness in Non-Repudiation Protocols," J. Software, 2006.
[10] J. Wen, M. Zhang, and H. Zhang, "Formal analysis of electronic payment protocols based on game logic," Microelectronics Computer, 2007.
[11] L. Chen, "New logic of analyzing electronic commerce security protocols," Computer Science, 2010.
[12] Y. Gao, D. Peng, and P. Tang, "A Formal Analysis Method for Optimistic Fair Exchange Protocol," J. Convergence Information Technology, 2013.
[13] K. Steve, "Formal analysis of optimistic fair exchange protocols," Universite Libre de Bruxelles, 2004.
[14] X. Xie, H. Zhang, "Fairness research of electronic commerce paying protocols based on finite automaton model," Computer Applications, 2004.
[15] H. Guo, Z. Li, L. Zhuang, and H. Ji, "New approach for analyzing of E-commerce protocol," Computer Science, 2010.
[16] W. Liu, S. Ma, Y. Si, and G. Hou, "A combining deterministic finite automaton with logic rules approach for analyzing of E-commerce protocol," J. Chinese Computer Systems, 2013.
[17] J. Dreier, A. Kassem, and L. Pascal, "Formal Analysis of E-Cash Protocols," International Conference on Security & Cryptography, vol.1, pp.65–75, 2015.
[18] M. Papa, O. Bremer, J. Hale, and S. Shenoi, "Formal Analysis of E-Commerce Protocols," The 5th International Symposium on Autonomous Decentralized Systems, 2001.
[19] C. Ouyang and J. Billington, "An Improved Formal Specification of the Internet Open Trading Protocol," 2004 ACM symposium on Applied computing, pp.779–783, 2004.
[20] G. Bella, F. Massacci, and L.C. Paulson, "Verifying the SET Purchase Protocols," J. Automated Reasoning, vol.36, no.1-2, pp.5–37, 2006.
[21] J.D. Guttman, "State and Progress in Strand Spaces: Proving Fair Exchange," J. Automated Reasoning, vol.48, no.2, pp.159–195, 2012.
[22] Y. Guo, L. Ding, Y. Zhou, and L. Guo, "Analysis for E-commerce protocols based on ProVerif," J. Communications, 2009.
[23] Y. Liu, X.-T. Liu, and C.-J. Tang, "A Novel Logic for Analyzing Electronic Payment Protocols," 3rd Annual International Conference on Information Technology and Applications, vol.7, 2016.
[24] M. Sirbu and J.D. Tygar, "NetBill: an internet commerce system optimized for network delivered services," IEEE Pers. Commun., vol.2, no.4, pp.34–39, Aug. 1995.
[25] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Trans. Inf. Theory, vol.22, no.6, pp.644–654, Nov. 1976.

Yi Liu received his bachelor's and master's degrees in National University of Defense Technology, China, in 2011 and 2014, respectively. He is a Ph.D. candidate in National University of Defense Technology. His main research interests are information security, electronic payment protocol analysis and blockchain technology.

Qingkun Meng received the B.S. and M.S. degrees in National University of Defense Technology, China, in 2010 and 2013, respectively. He is a doctoral candidate in National University of Defense Technology. His main research interests are information security, vulnerability detecting and machine learning.

Xingtong Liu received his Ph.D. degree from National University of Defense Technology, China, in 2014. He is a lecturer in National University of Defense Technology. His main research interests are information network security, quantum communication and protocol analysis.

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.