Fairness analysis of e-commerce protocols based on strand spaces

Yichun Liu
Guangdong Key Lab of Electronic Commerce, Guangdong University of Business Studies, Guangzhou, China
Email: liuyichun@126.com

Abstract: Strand space logic is a formal method for analysing the security protocol. The electronic commerce protocols are of more complex structures, for example, branch structures, or a protocol is composite of multiple sub-protocols, so the analysis of electronic commerce protocols is far more complex than the analysis of authentication protocols. Fairness is a very important feature in e-commerce protocol. But traditional belief logic is not suitable for this. In this paper, we analyse the strand and bundle of ISI payment protocol and then prove its unfairness. We also present a strand node path method to analyse ASW protocol, which consists of multiple sub-protocols with branch structure, and the strand space analysis shows that this protocol is fair.

Keywords: strand space; electronic commerce protocol; fairness; strand node path.


Biographical notes: Yichun Liu works as a researcher at Guangdong Key Laboratory of Electronic Commerce Market Application Technology in Guangzhou, P.R. China. His main research interests include information security and electronic commerce technology. He is a senior member of China Computer Federation and a senior member of Chinese Association for Cryptologic Research.

1 Introduction

The growing popularity of electronic commerce has resulted in the development of a number of electronic commerce protocols. The fairness is a primary and basic property of electronic commerce protocol, but the fairness of protocol is analysed manually in existing researches on electronic commerce, instead of being analysed formally.

With its strict mathematics foundation and broad adaptability, the formal analysis represents the direction of electronic commerce analysis. As yet, most of formal analysis was focused on authentication protocol and key exchange protocol.

Authentication protocols and key exchange protocols are of simple and sequential structure. But the electronic commerce protocols are of more complex structures, for example, if-then branch structures, or a protocol is composite of multiple sub-protocols, so that the analysis of electronic commerce protocols is far more complex than the analysis of authentication protocols and key exchange protocols. It is necessary that the suitable formal methodology be selected to analyse electronic commerce protocols effectively.

In this paper we address the problem of protocol verification using an existing logic verification technique known as strand space.

The paper is organised as follows: in the following section, we give an overview of related work in this field. The strand space logic is introduced in Section 3. Two typical electronic commerce protocols, ISI payment protocol (Medvinsky and Neuman, 1993) and ASW protocol (Asokan et al., 1997), are analysed by strand space logic in Section 4. Conclusions are drawn in Section 5.

2 Related work

Nowadays, most of researches on formal analyses of security protocol focus on authentication protocols, and there are seldom researches on electronic commerce protocol.

Many formal logic procedures have been proposed for formal analyses of authentication protocol, for example BAN logic (Burrows et al., 1990). BAN’s protocol goals can be expressed in terms of the beliefs that should ensue for protocol participants at the end of the protocol run. As a belief logic, BAN logic is not suitable for proving fairness of electronic commerce protocol, because belief logic examines if a principal believes a statement, but the proof of protocol fairness needs prove that no one take advantage over the other in electronic transaction.

Kailar (1996) presented a formal logic for analysing the accountability of electronic commerce protocol. Kailar logic emphasises on the accountability of transaction participants for other parties. As logic based on illation, Kailar’s logic
can be used to analyse the accountability and non-repudiation of electronic commerce protocol, but it cannot be used to analyse the fairness of protocol (Zhou et al., 1999).

Ryan and Schneider (2000) suggested analysing security protocol by CSP method and model checking tool FDR. CSP based on process algebra constructs and builds a finite state machine model of the protocol, and then analyses the protocol security by FDR tool. The state-checking tool searches possible state space and tries to find the possible attack paths. In 1996, Lowe (1996) used CSP and FDR model checker to analyse the NSPK authentication protocol and succeeded in finding a previously unpublished error in the protocol. Otherwise, Heintze et al. (1996) used CSP to check the atomicity of netbill and digitalcash payment system.

Thayer et al. (1998a, 1988b, 1999) combine NRL protocol analyser, CSP model checking technology and Paulson’s induction and proposed strand space logic in 1998, which bases on message algebra. Strand space theory utilises graph theory to describe protocol and attacks laconically. Strand space theory uses the concept ‘strand’ to describe the actions that participants send and receive messages, and the strands of various parties compose the strand space of a protocol, which represents the execution of a protocol. Message algebra defines the data structure of strand space and the relationship among various data items. Strand space theory brings new hope to formal analysis of security protocol.

3 Strand space theory

Strand space method is a new technology that mixes theorem proving and trace analysis. A strand is a sequence of events that protocol principals can execute. A strand of an honesty principal is made up of sent events and received events, and an attacker strand describes the actions of an attacker. A strand space is a set of strands consisting of strands for the various legitimate protocol parties, together with penetrator strands.

The kernel of strand space model is bundle structure. A bundle consists of a number of strands – legitimate or with penetrator strands. Strand space method is a new technology that mixes strand space and the relationship among various data items. Strand space theory uses the concept ‘strand’ to describe the actions that participants send and receive messages, and the strands of various parties compose the strand space of a protocol, which represents the execution of a protocol. Message algebra defines the data structure of strand space and the relationship among various data items. Strand space theory brings new hope to formal analysis of security protocol.

3.1 Strand space

An electronic commerce protocol usually relates to hash, encryption, decryption and signature operation. In original message algebra, only atomic items, join operation and encryption operation are described, and some new data type and data operation should be introduced into.

Definition 1: A set A is called as term space, if its elements are the possible messages that can be exchanged between principals in a protocol. We will refer to the elements of A as terms. An item space can be specialised by following sets:

- A set T of texts (representing the atomic messages).
- A set K of cryptographic keys disjoint from T.

Four binary operators:

- \( encr : K \times A \rightarrow A \)
- \( decr : K \times A \rightarrow A \)
- \( join : A \times A \rightarrow A \)
- \( sign : K \times A \rightarrow A \)

Two unary operators:

- \( hash : A \rightarrow A \)
- \( key : A \rightarrow K \)

As usual, for \( a,b,m \in A, k \in K \), we will write \( inv(k) \) as \( k^1 \), \( encr[k,m] \) as \( \{m\}^1, decr[k,m] \) as \( \{m\}^{-1}, sign[k,m] \) as \( \{m\} \), \( hash(a) \) as \( H(a) \), and \( join(a,b) \) as \( ab \).

The items \( \{m\}, H(m), \{m\} \) are called operation items. The elements and operations of the set \( T \cup K \) are called atomic items. The simple items include atomic items and operation items.

Definition 2: For the items \( h, g, g', g'' \in A, k \in K, h \) is the sub-item of item \( g \), namely \( h \subset g \), if one of the following is satisfied:

- \( h, g \in T \cup K \) and \( h = g \);
- if \( g = (g', g'') \), and \( h \subset g' \) or \( h \subset g'' \) or \( h = g \);
- if \( g = \{g\}'b \) and \( h \subset g' \) or \( h = g \).

Obviously, the sub-term relation \( \subset \) is a transitive and reflexive relation. In the structure of term algebra, the action of protocol participants is represented as a finite actions sequence of sending and receiving the messages, i.e. a strand \( s \). A strand can be denoted as a sequence \((<\sigma_1, a_1>, ..., <\sigma_i, a_i>)\), here \( <\sigma_i, a_i> \) is \( i \)-th signed term of strand \( s \), and \( \sigma_i \in \{+, -\} \) means sending or receiving messages, and \( a_i \) is an unsigned term.

Definition 3: A strand space is a set \( \Sigma \) with a trace mapping \( tr : \Sigma \rightarrow (2A)^* \)

Definition 4: Fix a strand space \( \Sigma \) and a strand \( s \in \Sigma \).

- A node is a pair \( <s, i> \), with \( s \in \Sigma \) and \( i \) an integer satisfying \( 1 \leq i \leq \text{length}(tr(s)) \). The set node is denoted by \( N \). We will say the node \( <s, i> \) belongs to the strand \( s \).
- If \( n = s, i > N \), then \( \text{index}(n) = i \) and \( \text{strand}(n) = s \). Define \( \text{term}(n) \) to be \( (tr(s))_i \), i.e. the \( i \)-th signed term in
the trace of s. Similarly, uns_term(n) is \((\tau(r(s)))_n\), i.e. the
unsigned part of the \(i\)-th signed term in the trace of s.

- If \(n_1, n_2 \in N\), \(n_1 \rightarrow n_2\) means term\(n_1\) = \(+a\) and
term\(n_2\) = \(-a\). It means that node \(n_1\) sends the message \(a\),
which is received by \(n_2\), creating a causal link between their strands.

- If \(n_1, n_2 \in N\), then \(n_1 \Rightarrow n_2\) means \(n_1, n_2\) occur on the
same strand with index\(n_1\) = index\(n_2\). It expresses that \(n_1\) is an immediate causal predecessor of \(n_2\) on the strand.

\(N\) becomes an ordered graph with both sets of edges \(n_1 \rightarrow n_2\)
and \(n_1 \Rightarrow n_2\). The basic structure of a strand space of a
protocol can be represented by a graph \((N, \rightarrow \cup \Rightarrow))\), and
the implementation of the protocol can be described formally by the graph.

Definition 5: Let \(B\) be a set of edges, and let \(N_B\) be the set of
nodes incident with any edge in \(B\). \(B\) is a bundle if:

- \(B\) is finite.

- If \(n_1 \in N_B\) and term \((n_1)\) is negative, then there is a
unique \(n_2\) such that \(n_2 \rightarrow n_1\) \(\in B\).

- If \(n_1 \in N_B\) and \(n_2 \Rightarrow n_1\) then \(n_2 \Rightarrow n_1\) \(\in B\).

- \(B\) is acyclic.

A node \(n\) is in a bundle \(B\), written \(n \in B\), if \(N_B\) is the node
set of bundle \(B\) and \(n \in N_B\). A strand \(s\) is in a bundle \(B\),
written \(s \in B\), if all of its nodes are in \(N_B\).

There are two important free assumptions in strand
space theory as follows:

Axiom 1: \(\forall t_1, t_2 \in A, \forall k_1, k_2 \in K\), the following
statement is inferred:

\[\{t_1\}_{k_1} \cap \{t_2\}_{k_2} \Rightarrow t_1 = t_2, \quad k_1 = k_2\]

Axiom 2: \(\forall t_1, t_2, t_3, t_4 \in A, \forall k \in K\), the following
statements are inferred:

- \((t_1, t_2) = (t_1, t_3) \Rightarrow t_1 = t_3, t_2 = t_4\)
- \((t_1, t_2) \neq \{t_1\}_k\)
- \((t_1, t_2) \notin T \cup K\)
- \(\{t_1\}_k \notin T \cup K\)

The following free assumption should be supplemented to
analyse the protocols that contain decryption operation.

Axiom 3: \(\forall t \in A, \forall k \in K\), the following statement is
inferred:

\[\{\{t\}\}_k \}\)_{k+1} = t\]

The above axioms describe the message algebra about the
set \(T\) and \(K\) as well as operations encr, decr and join.

3.2 Fairness on strand space theory

A common electronic payment protocol includes three roles:
a buyer, a seller and a trusted third party. The strands
corresponding to these roles are a buyer strand, a seller
strand and a trusted party strand.

The fairness of electronic commerce protocol can be
described by strand space logic as follows:

Definition 6: An exchange protocol between party \(X\) and
party \(Y\) is said fair, if the following holds:

1. If party \(X\) is honesty and there are no negative nodes
containing a sub-terms that represent \(X\)'s expected
items or an affidavit for the items from trusted party in
\(X\)'s strand, there are no negative nodes containing the
sub-items that represent \(Y\)'s expected items in \(Y\)'s
strand.

2. If party \(X\) is honesty and there are negative nodes
containing a sub-terms that represent \(X\)'s expected
items or an affidavit for these items from trusted party in
\(X\)'s strand, there are negative nodes containing the
sub-items that represent \(Y\)'s expected items or an
affidavit for the items from trusted party in
\(Y\)'s strand.

To analysis the fairness of an electronic payment protocol,
the strands and bundles should firstly be expressed, and then
possible node path should be listed. The analysis of every
node paths can judge the fairness of the protocol. If the
implementations along every node path are fair, then the
protocol is regarded as a fair protocol. If a party has the
advantage over another when a protocol is implemented
along a strand node path, then the protocol is not fair.

4 Case analysis

4.1 ISI payment protocol

4.1.1 Protocol description

Southern California University's ISI payment protocol is
intended for offering real time electronic payments with
provision of anonymity over an insecure network (Medvinsky
and Neuman, 1993). The protocol is articulated below:

1. \(A \rightarrow B\): \(K_{ab}\)

2. \(B \rightarrow A\): \(\{K_{b}\}K_{ab}\)

3. \(A \rightarrow B\): \(\{\{coins\}_{K_{ab}}, SK_{a}, K_{ses}, s_id\}_{b}\)

4. \(B \rightarrow CS\): \(\{\{coins\}_{K_{ab}}, SK_{b}, transaction\}_{K_{cs}}\)

5. \(CS \rightarrow B\): \(\{new\_coins\}_{K_{cs}}SK_{b}\)

6. \(B \rightarrow A\): \(\{amount, Tid, date\}_{K_{s}}\}_{SK_{a}}\)

\(K_{ab}\) is the session key between \(A\) and \(B\). \(K_{a}\) is \(A\)'s public key
and \(K_{b}\) is \(B\)'s public key. \(SK_{a}\) is \(A\)'s session key and \(SK_{b}\) is \(B\)'s
session key. \(K_{ses}\) is the session key for the service and \(s_id\)
is the identifier of the service. \(A\) pay \(B\) the electronic coins
\(coins\), and the currency server \(CS\) pay \(B\) the cash \(new\_coins\).
The receipt of payment is \(\{amount, Tid, date\}_{K_{s}}\).
In step 1 and 2, A obtains B’s public key. In step 3, A encrypts the electronic payment $coins$, the service identifier $s_id$ and the key $K_ses$ with B’s public key, and then sends them to $B$. In step 4, $B$ verifies the validity of the payment, and sends it to currency server $CS$. In step 5, $CS$ verifies the validity of the payment $coins$, and pays $B$. In step 6, $B$ sends the receipt $\{amount, Tid, date\}_{K_b}^{-1}$ to $A$.

ISI payment protocol is fair if $B$ can obtain valid payment and $A$ can obtain payment receipt, or $A$ cannot obtain payment receipt while $B$ cannot obtain valid payment.

### 4.1.2 Strand space analysis

The ISL protocol is described in Figure 1.

**Figure 1** ISI payment protocol

```
A ----------> CS ----------> B

K_a

\{(coins)_{K_a}^{-1}, SK_a, K_ses, s_id\}_{K_b}

\{(coins)_{K_a}^{-1}, SK_a, K_ses, s_id\}_{K_b}

\{(coins)_{K_a}^{-1}, SK_a, K_ses, s_id\}_{K_b}

\{(amount, Tid, date)_{K_b}^{-1}\}_{S_b}
```

The protocol may be finished when both $A$ and $B$ execute protocol honestly, or the protocol is aborted when one party misbehaves or communication failure occurs. When the protocol is finished honestly, it has maximum strand nodes. The strand nodes set of a protocol that is aborted abnormally is a subset of the strand nodes set of a protocol that is finished normally.

The maximum nodes set of $A$’s payer strand $s_1$ is as follows:

- $<s_{1,1}>$: $+K_{sub}$
- $<s_{1,2}>$: $-\{K_b\}_{K_{ab}}$
- $<s_{1,3}>$: $+\{coins\}_{K_a}^{-1}, SK_a, K_ses, s_id\}_{K_b}$
- $<s_{1,4}>$: $-\{amount, Tid, date\}_{K_b}^{-1}\}_{S_b}$

The possible originator protocol implementation strands include:

1. $<+K_{sub} - \{K_b\}_{K_{ab}} + \{coins\}_{K_a}^{-1}, SK_a, K_ses, s_id\}_{K_b} - \{amount, Tid, date\}_{K_b}^{-1}\}_{S_b}$
2. $<+K_{sub} - \{K_b\}_{K_{ab}} + \{coins\}_{K_a}^{-1}, SK_a, K_ses, s_id\}_{K_b}$
3. $<+K_{sub} - \{K_b\}_{K_{ab}}$
4. $<+K_{sub}>$

In case 2, the node $<s_{1,3}>$ of $A$’s strand contains a sub-term denoting valid payment for $B$ but there are no nodes denoting receipt from $B$ in $A$’s strand, i.e. there is the possibility that $A$ has sent the valid payment but $A$ has not obtained valid payment receipt. So ISI protocol is not fair.

### 4.2 ASW protocol

#### 4.2.1 Protocol description

Asokan et al. (1997) presented a famous optimistic protocol for generic exchange, called ASW protocol. A two-party ASW protocol exchange electronic goods between two participants, $O$ and $R$, and $T$ is trusted party. In order to start an exchange, each party $X$ (one of $O$ or $R$) has to input the following parameters:

1. $item_X$: the item that $X$ wants to send.
2. $descr_X$: a description of item $X$.
3. $expect_X(descr_X, descr_Y)$: it evaluates to true if the user $X$ is satisfied when receiving an item described by $descr_Y$ in exchange for an item described by $descr_X$.
4. $fits(descr_X, item_Y)$: a predicate that evaluates to true if the description fits the item.

ASW protocol consists of three sub-protocols: the exchange sub-protocol, recovery sub-protocol for $O$ and recovery sub-protocol for $R$. Figure 2 depicts the exchange sub-protocol. The basic idea of the protocol is that the originator $O$ and the recipient $R$ start by promising each other an exchange of items. If they do not agree on the exchange the protocol is aborted. Otherwise they proceed to exchange the items. If no exception occurs, the protocol only consists of the exchange sub-protocol and does not involve $T$. If disputes occur, $O$, $R$, and $T$ start an error-recovery phase. Recovery initiated by $O$ is depicted in Figure 3. Recovery initiated by $R$ is depicted in Figure 4.

#### 4.2.2 Strand space analysis

ASW protocol consists of multiple sub-protocols with if-then branch structure. We present a new method based on strand node path to analyse the fairness of the protocol.

**Figure 2** Exchange sub-protocol of optimistic exchange protocol

```
\begin{align*}
&\text{if } item_X, descr_X, expect_X(\cdot) \text{ \&\& } fits descr_X, item_Y \text{ then } m_1 = m(\cdot, descr_X, item_Y) \\
&\text{else } [\text{Recovery for } O] \\
&\text{if } item_X, descr_Y, expect_Y(\cdot) \text{ \&\& } fits descr_Y, item_X \text{ then } m_2 = m(\cdot, descr_Y, item_X) \\
&\text{else } [\text{Recovery for } R] \\
\end{align*}
```
Let $s_1$ be $O$’s strand and $s_2$ be $R$’s strand in exchange sub-protocol, $s_1'$ is $O$’s strand and $s_2'$ is $R$’s strand in recovery sub-protocol for $O$, and $s_1''$ is $O$’s strand and $s_2''$ is $R$’s strand in recovery sub-protocol for $R$.

The strand nodes of $O$ and $R$ in the payment protocol are shown in Table 1.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Message items</th>
<th>Nodes</th>
<th>Message items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt;s_1,1&gt;$</td>
<td>$+m_1$</td>
<td>$&lt;s_1',1&gt;$</td>
<td>$-m_1$</td>
</tr>
<tr>
<td>$&lt;s_1,2&gt;$</td>
<td>$-m_2$</td>
<td>$&lt;s_2',2&gt;$</td>
<td>$+m_2$</td>
</tr>
<tr>
<td>$&lt;s_1,3&gt;$</td>
<td>$(item_{oA}, key_{oA})$</td>
<td>$&lt;s_2,3&gt;$</td>
<td>$-(item_{oA}, key_{oA})$</td>
</tr>
<tr>
<td>$&lt;s_1,4&gt;$</td>
<td>$-(item_{oR}, r_{oA}, key_{oA})$</td>
<td>$&lt;s_3,4&gt;$</td>
<td>$(item_{oR}, r_{oA}, key_{oA})$</td>
</tr>
<tr>
<td>$&lt;s_1,5&gt;$</td>
<td>$+r_{oA}$</td>
<td>$&lt;s_2,5&gt;$</td>
<td>$-r_{oA}$</td>
</tr>
<tr>
<td>$&lt;s_1',1&gt;$</td>
<td>$(m_1, m_2, y_{oA})$</td>
<td>$&lt;s_1',5&gt;$</td>
<td>$-$(item$<em>{oA}, key</em>{oA})$</td>
</tr>
<tr>
<td>$&lt;s_1',2&gt;$</td>
<td>$+$(item$<em>{oA}, key</em>{oA})$</td>
<td>$&lt;s_2',2&gt;$</td>
<td>$(item_{oR}, r_{oA}, key_{oA})$</td>
</tr>
<tr>
<td>$&lt;s_1',3&gt;$</td>
<td>$-(item_{oR}, r_{oA}, key_{oA})$</td>
<td>$&lt;s_3',3&gt;$</td>
<td>$-r_{oA}$</td>
</tr>
<tr>
<td>$&lt;s_1',4&gt;$</td>
<td>$+r_{oA}$</td>
<td>$&lt;s_2',4&gt;$</td>
<td>$(item_{oR}, r_{oA}, key_{oA})$</td>
</tr>
<tr>
<td>$&lt;s_1'',5&gt;$</td>
<td>$-$(item$<em>{oA}, key</em>{oA})$</td>
<td>$&lt;s_3'',4&gt;$</td>
<td>$-$(item$<em>{oA}, key</em>{oA})$</td>
</tr>
</tbody>
</table>

The possible strand node paths of an honest originator are shown in Figure 5, and the possible strand node paths of an honest responder are shown in Figure 6. Both originator and responder have multiple strand node paths.

When the originator $O$ executes the protocol honestly, its possible strand node paths include:

Path 1: $<s_1,1>$
Path 2: $<s_1,1>, <s_1,2>$
Path 3: $<s_1,1>, <s_1,2>, <s_1,3>, <s_1,4>, <s_1,5>$
Path 4: $<s_1,1>, <s_1,2>, <s_1,3>, (<s_1',1), (<s_1',2), (<s_1',4), (<s_1'',4)$
Path 5: $<s_1,1>, <s_1,2>, <s_1,3>, (<s_1',1), (<s_1',2), (<s_1',5)$

In path 1 and path 2, no nodes contain the sub-item $item_{oA}$, i.e. $O$ has not sent any goods.

In path 3, the node ($s_1,3$) contains item $(item_{oA}, key_{oA})$, the node ($s_1,5$) contains item $r_{oA}$, and they are positive, while the node ($s_1,4$) contains item $(item_{oR}, r_{oA}, key_{oA})$ and it is negative.

In path 4, the node ($s_1,3$) contains sub-item $(item_{oA}, key_{oA})$, the node ($s_1,4$) contains item $r_{oA}$, and they are positive, while the node ($s_1,5$) contains item $(item_{oR}, r_{oA}, key_{oA})$ and it is negative.

In path 5, the node ($s_1,3$) contains item $(item_{oA}, key_{oA})$ and it is positive, while the node ($s_1',5$) contains item $sig_{oA}(m_1, m_2, y_{oA})$ that is used for affidavit and it is negative.

The above analysis shows that the originator $O$ can obtain the item $(item_{oA}, r_{oA}, key_{oA})$ from $R$ or the affidavit item $sig_{oA}(m_1, m_2, y_{oA})$ if $O$ has sent the item $(item_{oA}, key_{oA})$ to $R$.

When the responder $R$ executes the protocol honestly, its possible strand node paths include:

Path 1: $<s_2,1>$
Path 2: $<s_2,1>, <s_2,2>$
Path 3: $<s_2,1>, <s_2,2>, <s_2,3>$
Path 4: $<s_2,1>, <s_2,2>, <s_2,3>, <s_2,4>, <s_2,5>$
Path 5: $<s_2,1>, <s_2,2>, <s_2,3>, <s_2,4>, (<s_2',1), (<s_2',2), (<s_2'',3)$
Path 6: $<s_2,1>, <s_2,2>, <s_2,3>, <s_2,4>, (<s_2',1), (<s_2'',2), (<s_2'',4)$

In path 1, path 2 and path 3, no nodes contain the sub-item $(item_{oA}, key_{oA})$, i.e. $R$ has not sent any goods.

In path 4, the node ($s_2,4$) contains item $(item_{oA}, r_{oA}, key_{oA})$ and it is positive, while the node ($s_2,3$) contains item $(item_{oA}, key_{oA})$ and the node ($s_2,5$) contains the item $o$ and they are negative.
In path 5, the node \((s_2,4)\) contains sub-item \((item_b, r_b, key_b)\) and it is positive, while the node \((s_2,3)\) contains item \((item_o, key_o)\) and the node \((s_2',3)\) contains the item \(o\) and they are negative.

In path 6, the node \((s_2,4)\) contains sub-item \((item_b, r_b, key_b)\) and it is positive, while the node \((s_2,3)\) contains item \((item_o, key_o)\) and sub-item \(\text{`sig}_A(m_1, m_2, y_R)\)’ that is used for affidavit and it is negative.

The above analysis shows that the responder \(R\) can obtain the item \((item_o, key_o, r_o)\) from \(O\) or the affidavit item \(\text{`sig}_A(m_1, m_2, y_R)\) if \(R\) has sent the item \((item_b, r_b, key_b)\) to \(O\).

The analysis of the strand node path show that each party can obtain his expected item or the affidavit from the trust party when it has sent his items, and no one are at disadvantage, i.e. the optimistic protocol is fair.

5 Conclusion

Existing formal analyses are focused on authentication protocols and key exchange protocols, and these protocols are usually simple. The electronic commerce protocols have more complex structures, for example branch structures, or a protocol is a composite of multiple sub-protocols, and it is difficult to analyse the electronic commerce protocol by traditional belief logic. Kailar logic is suitable for analysing the accountability of commerce protocol and it is not suitable for analysis of fairness. By introducing strand node path method, the electronic protocol with complex structure can be analysed effectively with strand space logic. In this paper, ISI payment protocol is proven unfair and ASW protocol is proven fair by strand space analysis. Our paper has given a generic method for analysing the electronic commerce protocol.

Furthermore, by applying strand node path techniques, our proposed method can be flexibly modified for verification of universal electronic protocols, such as abuse-free electronic payment protocol (Fan et al., 2012).

Acknowledgements

This work was supported by the MOE (Ministry of Education in China) Project of Humanities and Social Sciences (NO. 12YJAZH079), the Innovation Teams Fund of Guangdong University of Business Studies, and the Natural Science Foundation of Guangdong Province, China (NO. S2011010001581).

References


