Modelling and studying the misbehaviour in a fair-exchange e-commerce security protocol using PEPA

Ohud Almutairi$^1$ and Nigel Thomas$^1$

Newcastle University, Newcastle upon Tyne, UK
{o.m.m.almutairi2,nigel.thomas}@newcastle.ac.uk

Abstract. This paper explores the performance cost introduced by a security protocol known as an anonymous and failure resilient fair-exchange e-commerce protocol. The protocol guarantees customer anonymity and fair exchange between two parties in an e-commerce environment. In this paper, the protocol is studied and modelled when misbehaviour between participants occur. Models are formulated using the PEPA formalism to investigate the performance overheads introduced by the security properties and behaviour of the protocol when the dispute between the parties exists. This study uses a PEPA Eclipse plug-in to support the creation and evaluation of the proposed PEPA models.

Keywords: PEPA · Security protocol · Misbehaviour.

1 Introduction

Computing systems are becoming increasingly complex and consist of multiple interactive components. Performance has been seen as an important aspect for evaluating computing systems. Many computing systems are connected to the network, either privately or publicly, and this can pose vulnerability issues, exposing systems to threats and attacks. Security protocol can add an extra overhead to a system, directly influencing its performance; this is a problem for many different domains. For example, the performance of a web server reduces in response to the implementation of the secure sockets layer protocol [3, 2].

Performance and security are essential aspects for almost all systems. Thus, it is important to develop a system affording an optimal balance between security and performance concerns. Therefore, the extra cost that security aspects contribute to the system’s performance has attracted widespread attention. As a result developers have conducted explorations and taken measurements with the aim of developing a secure system that gives a satisfactory performance[3, 8, 9].

This paper will study and explore a type of non-repudiation e-commerce security protocols when misbehaviour and dispute between the customer and merchant occurs. This protocol is called an anonymous and failure resilient fair-exchange e-commerce protocol. It was proposed by Ray et al. [6]. It guarantees
a fair-exchange between two parties and satisfies the following features: first, fairness – no party can have any advantages over the other party during the exchange course; second, the anonymity of the parties – a customer can interact without disclosing any personal information; third, no manual dispute resolution; fourth, not relying on the service of a single trusted third party (TTP) – instead, multiple TTPs are available to provide services; fifth, offline TTP – the involvement of such a party must be at a minimum level, only when any problem occurs; and finally, any types of digital merchandise can be exchanged. Moreover, the protocol is based on an approach called ‘cross-validation’, which allows the customer to validate the encrypted electronic product without decrypting it.

Based on the description provided by Ray et al in [6], the protocol has two versions: with and without an anonymity feature. In this paper, we considered the protocol version that ensures customer privacy is protected from any other parties. The customer does not need to share any personal information with a merchant to buy. Thus, the customer’s true identity is hidden from the merchant. Ray et al. modified the basic failure resilient fair-exchange protocol to prevent the customer’s personal information from being known by the merchant by following the electronic cash system described in [5]. In the basic version, the payment token that the customer sends to the merchant contains some personal information, such as the customer’s identity and bank account information. Therefore, the merchant will have detailed personal information of the customer once it receives the payment token. However, in this modified version, the customer uses digital base money to buy from merchants. By using this method, merchants cannot obtain any personal information from the customer or create a customer profile without permission.

The protocol relies on TTPs but does not need them to be active at any time except if a problem occurs. Therefore, the protocol has two main descriptions depending on the type of TTP involvement: offline TTP (basic) and online TTP (extension) [6]. With offline TTP involvement type, there is no TTP active involvement as no parties misbehave or prematurely terminate the protocol. However, with online TTP, when parties misbehave or prematurely terminate the protocol, the TTP must be involved in resolving the problem and ensuring fair-exchange. In [1], we studied the protocol’s performance without dispute between parties. In this paper, we study the protocol’s performance when the dispute between the parties occurs. In addition, the discussion focuses on the behaviour aspects of the protocols in order to analyse their performance.

The approach is used to model the protocol is Performance Evaluation Process Algebra (PEPA). PEPA is a well-known implementation of Stochastic Process Algebras (SPA). A system is modelled in PEPA formalism as a set of components which interact and engage individually or with other components in activities in order to evaluate its performance [4]. Thus, the components represent the active parts in the system and the behaviour of each part is represented by its activities. The creation and performance evaluation of a PEPA model is supported by the PEPA Eclipse plug-in [7]. This tool has been developed to support the Markovian steady-state analysis, Stochastic Simulation Algorithms
(SSA) analysis, and Ordinary Differential Equations (ODE) analysis of PEPA models in the Eclipse Platform [7].

The paper is organized as follows. Section 2 provides the protocol specification. In Section 3, the proposed PEPA model of the protocol is presented. Section 4 presents the protocol’s evaluation and results. Finally, Section 5 concludes the paper by providing an overview of the study findings and future work.

2 Protocol specification

2.1 An anonymous and failure resilient fair-exchange e-commerce protocol

This subsection provides an informal description of the protocol. The formal description of the protocol and security-related details are provided in [6]. Before the protocol is initiated, the environment needs to be set up with the same steps detailed in [6, 1]. The customer (C) uses a pseudo identifier C’ when starting a new transaction with the merchant (M) to preserve the anonymity of C. Thus, no parties in the protocol except the customers themselves have sufficient information to link the C’ used in the transaction with C, which is the real customer identity. The following are the main nine interaction steps. A more detailed description of the steps are provided in [6, 1]. All texts in bold indicate the names of the actions used in the PEPA model presented in this paper:

1. download (TTP ⇒ C): C visits the TTP website and downloads the encrypted electronic product from the TTP server. This encrypted electronic product can be used to validate the product received from M.
2. requestBDigitalCoins (C ⇒ B): C sends a request to B for digital coins.
3. sendCDigitalCoins (B ⇒ C): B sends the signed blinded coin to C.

Fig. 1: An anonymous and failure resilient fair-exchange e-commerce protocol.
4. sendMPO (C' ⇒ M): C sends a message containing the purchase order (PO) to M.
5. sendCEP or sendCAbort (M ⇒ C'): M sends the encrypted product to C' or sends an abort statement to end the transaction if M is not satisfied.
6. sendMCoinDk or sendMAbort (C' ⇒ M): C' sends the decryption key of the digital coin to M or sends an abort message to end the transaction. If M sends the encrypted electronic product, then C' validates it with the encrypted electronic product received from TTP (step 1). If the product is valid, C' sends the decryption key for the digital coin to M and then waits for the product decryption key by setting a timer. If C' does not receive the key within the time set, they will require TTP involvement. If the product is not valid, C' sends an abort statement to M.
7. sendBCoinByM or sendCAbort (M ⇒ B or M ⇒ C'): M sends B the digital coin for validation, or M sends C' an abort message to terminate the transaction. If M is unsatisfied for any reason, M sends C' an abort message to terminate the transaction.
8. sendMyes or sendMno (B ⇒ M): B sends M either ‘yes’ or ‘no’. If the coin has been spent, B sends M ‘no’. If the coin has not been spent, B credits M’s account with the same amount of money as the digital coin and then sends M ‘yes’.
9. sendCPDk or sendCAbort (M ⇒ C'): M sends the electronic product decryption key to C' after receiving ‘yes’ from B, or ends the transaction by sending an abort message to C' after receiving ‘no’ from B.

2.2 The extended protocol for handling misbehaviours and communication problems

This subsection presents scenarios to solve any dispute between the merchant and the customer. When misbehaviours and/or communication problems occur, the extended protocol is initiated, and TTP status is changed to online during the protocol execution. The execution of the extended protocol is started when the customer’s timer expires (after Step 6 in the protocol), and the protocol does not reach completion status or when the customer receives an abort message or an invalid product decryption key in Step 9.

A dispute resolution is initiated when a customer sends TTP an initiation message that contains an evidence of misbehaving. The misbehaviour scenarios solved by the extended protocol are illustrated as follows (all texts in bold indicate the names of the actions used in the PEPA models presented in this paper):

Merchant behaves improperly

Scenario 1 : M sends an invalid product decryption key (Step 9 in the protocol steps). The interaction actions for resolving this dispute are as follows:
1. **seekingHelpFromTTP** and then **sendTTPinfo**: C initiates the execution of the extended protocol by seeking help from TTP. Then C sends an initiation message to TTP to seek a resolution of the problem.

2. **validateCoinToB**: TTP receives the customer’s dispute resolution request. Then it starts to contact B to validate the coin.

3. **sendTTPyes**: B confirms that the coin is valid, which means that the customer played fairly.

4. **askMForValidK**: TTP orders M to send a valid product decryption key.

5. **sendTTPvalidK** or **timeoutTTP**: M responds within the timeout period by sending the valid product decryption key to TTP, or M does not respond and the timeout expires.

6. **forwardKtoC** or **sendCkByTTP** and **takeActionAgainstM**: If TTP receives the valid product decryption key from M, it forwards it to C. However, if TTP does not receive the valid product decryption key within a specified timeout period, TTP sends C the preserved product decryption key and then takes action against M.

**Scenario 2**: M sends an invalid product decryption key (Step 9 in the protocol steps). The coin in this scenario is invalid. The interaction actions for resolving this dispute are as follows:

1. **seekingHelpFromTTP** and then **sendTTPinfo**: same as step 1 in the first scenario.

2. **validateCoinToB**: same as step 2 in the first scenario.

3. **sendTTPno**: B confirms that the coin is invalid, which means that TTP needs to investigate who spent the coin.

4. **investigationInvalidCoinToB**: TTP contacts B to investigate who spent the coin.

5. **mspentTheCoinToTTP**: B confirms that the coin is spent by M.

6. **askMForValidK**: same as step 4 in the first scenario.

7. **sendTTPvalidK** or **timeoutTTP**: same as step 5 in the first scenario.

8. **forwardKtoC** or **sendCkByTTP** and **takeActionAgainstM**: same as step 6 in the first scenario.

**Scenario 3**: M disappears without sending a valid product decryption key (Step 9 in the main protocol steps). The interaction actions for resolving this dispute are as follows:

1. **cTimeoutExpired** and then **sendTTPinfo**: C initiates the extended protocol by sending an initiation message after the timeout period for receiving the decryption key has expired.

2. **validateCoinToB**: same as step 2 in the first scenario.

3. **sendTTPyes**: same as step 3 in the first scenario.

4. **askMForValidK**: same as step 4 in the first scenario.

5. **sendTTPvalidK** or **timeoutTTP**: same as step 5 in the first scenario.

6. **forwardKtoC** or **sendCkByTTP** and **takeActionAgainstM**: same as step 6 in the first scenario.
Scenario 4: M claims that a valid product decryption key has not been sent because an invalid coin’s decryption key has been received from C. The interaction actions for resolving this dispute are as follows:

1. sendTTPReason: M responds to TTP by identifying the reason for not sending the valid product decryption key to C after TTP contacts it to send a valid decryption key.
2. sendTTPvalidK: M must still send TTP the valid product decryption key.
3. sendMpKbyTTP and forwardKtoC: When TTP receives the valid product decryption key from M, it sends M the valid coin decryption key and forwards the valid product decryption key to C.

Customer behaves improperly In this case, after C sends TTP an initiation message for the extended protocol, TTP starts contacting B to validate the coin. If the coin is invalid (sendTTPno), TTP then starts contacting B to investigate who spent the coin (investigationInvalidCoinToB). If B confirms that the customer is who spent the coin (cspentTheCoinToTTP), TTP will not forward the valid product decryption key to C (discoverMisbehavingC).

3 PEPA models of the extended protocol

This section presents the proposed PEPA model for the protocol extension for handling misbehaviours and communication problems. The model is an extended protocol to solve the misbehaving event between M and C parties with probabilities distributed of M misbehaving. The PEPA model comprises four main components. The four components are Merchant (M), Customer (C), Trust Third Party (TTP) and Bank (B). M, C and TTP are sequential components in the PEPA model, whereas B is a static component.

The extended PEPA model with the probabilities distribution of M misbehaviour is formulated as follows:

**Merchant component**

\[
M_0 \overset{def}{=} (sendMPO, r_{sendMPO}).M_1 \\
M_1 \overset{def}{=} (sendCEP, r_{sendCEP}).M_2 + (sendAbort, r_{sendAbort}).M_8 \\
M_2 \overset{def}{=} (sendMCoinDk, r_{sendMCoinDk}).M_3 + (sendMAbort, r_{sendMAbort}).M_6 \\
M_3 \overset{def}{=} (startContactB, r_{startContactB}).M_3a + (sendAbort, r_{sendAbort}).M_6 \\
M_{3a} \overset{def}{=} (sendBCoinByM, r_{sendBCoinByM}).M_4 \\
M_4 \overset{def}{=} (sendMyes, r_{sendMyes}).M_5 + (sendMno, r_{sendMno}).M_7 \\
M_5 \overset{def}{=} (sendCPDk, r_{sendCPDk}).M_6 + (cTimeoutExpired, r_{cTimeoutExpired}).M_6 \\
M_6 \overset{def}{=} (complete, r_{complete}).M_0 + (askMforValidK, r_{askMforValidK}).M_9 \\
M_7 \overset{def}{=} (sendAbort, r_{sendAbort}).M_6 \\
M_8 \overset{def}{=} (sendMAbort, r_{sendMAbort}).M_6 \\
M_9 \overset{def}{=} (sendTTPvalidK, r_{sendTTPvalidK}).M_6 \\
+ (timeoutTTP, r_{timeoutTTP}).M_{10} \\
+ (sendTTPReason, r_{sendTTPReason}).M_{11}
\]
The above model component specifies M's different behaviours, moving from $M_0$ to $M_{12}$. It has fourteen behaviours to reflect the protocol's steps for M. M moves sequentially between the different behaviours based on the activities specified in the PEPA component. The actions presented reflect the protocol's interaction steps related to M.

**Customer component**

\[ C_0 \overset{\text{def}}{=} (\text{download}, r_d).C_1 \]
\[ C_1 \overset{\text{def}}{=} (\text{requestBDigitalCoins}, r_{\text{requestBDc}}).C_2 \]
\[ C_2 \overset{\text{def}}{=} (\text{sendCDigitalCoins}, r_{\text{sendCdc}}).C_3 \]
\[ C_3 \overset{\text{def}}{=} (\text{sendMPO}, r_{\text{sendMPO}}).C_4 \]
\[ C_4 \overset{\text{def}}{=} (\text{sendCEP}, r_{\text{sendCEP}}).C_5 + (\text{sendCAbort}, r_{\text{sendCAbort}}).C_8 \]
\[ C_5 \overset{\text{def}}{=} (\text{sendMCOinDk}, r_{\text{sendMCOinDk}}).C_6 + (\text{sendMAbort}, r_{\text{sendMAbort}}).C_7 \]
\[ C_6 \overset{\text{def}}{=} (\text{sendCPDk}, p \cdot r_{\text{sendCPDk}}).C_7 + (\text{sendCPDk}, (1 - p) \cdot r_{\text{sendCPDk}}).C_9 \]
\[ + (\text{sendCAbort}, p \cdot r_{\text{sendCAbort}}).C_7 \]
\[ + (\text{sendCAbort}, (1 - p) \cdot r_{\text{sendCAbort}}).C_9 \]
\[ + (\text{cTimeoutExpired}, p \cdot r_{\text{cTimeoutExpired}}).C_7 \]
\[ + (\text{cTimeoutExpired}, (1 - p) \cdot r_{\text{cTimeoutExpired}}).C_9 \]
\[ C_7 \overset{\text{def}}{=} (\text{complete}, r_{\text{complete}}).C_0 \]
\[ C_8 \overset{\text{def}}{=} (\text{sendMAbort}, r_{\text{sendMAbort}}).C_7 \]
\[ C_9 \overset{\text{def}}{=} (\text{sendTTPinfo}, r_{\text{sendTTPinfo}}).C_{10} \]
\[ C_{10} \overset{\text{def}}{=} (\text{forwardKtoC}, r_{\text{forwardKtoC}}).C_7 + (\text{sendCkByTTP}, r_{\text{sendCkByTTP}}).C_7 \]
\[ + (\text{discoverMisbehavingC}, r_{\text{discoverMisbehavingC}}).C_7 \]

The above model component specifies C’s different behaviours, moving from $C_0$ to $C_{10}$. It has eleven behaviours to reflect the protocol’s steps for C. C moves sequentially between the different behaviours based on the activities specified in the PEPA component. The actions presented reflect the protocol’s interaction steps related to C. In $C_6$, we introduced a probability distributed of M to misbehave. $p$ is the probability of M to be honest and $(1 - p)$ is the probability of M to be misbehaving. Therefore, in $C_6$, there are 6 actions could happen either $\text{sendCPDk}$ at rate $r_{\text{sendCPDk}} \cdot p$ moving to $C_7$, $\text{sendCPDk}$ at rate $r_{\text{sendCPDk}} \cdot (1 - p)$ moving to $C_9$ to seek a dispute resolution form TTP, $\text{sendCAbort}$ at rate $r_{\text{sendCAbort}} \cdot p$ moving to $C_7$, $\text{sendCAbort}$ at rate $r_{\text{sendCAbort}} \cdot (1 - p)$ moving to $C_9$, $\text{cTimeoutExpired}$ at rate $r_{\text{cTimeoutExpired}} \cdot p$ moving to $C_7$ or $\text{cTimeoutExpired}$ at rate $r_{\text{cTimeoutExpired}} \cdot (1 - p)$ moving to $C_9$. 
TTP component

\[
\begin{align*}
&TTP_0 \overset{\text{def}}{=} (\text{download}, r_d).TTP_1 + (\text{sendTTPinfo}, r_{\text{sendTTPinfo}}).TTP_1 \\
&TTP_1 \overset{\text{def}}{=} (\text{validateCoinToB}, r_{\text{ve}}).TTP_2 \\
&TTP_2 \overset{\text{def}}{=} (\text{sendTTPyes}, r_{\text{yes}}).TTP_3 + (\text{sendTTPno}, r_{\text{no}}).TTP_7 \\
&TTP_3 \overset{\text{def}}{=} (\text{askMforValidK}, r_{\text{askMforValidK}}).TTP_4 \\
&TTP_4 \overset{\text{def}}{=} (\text{sendTTPvalidK}, r_{\text{sendTTPvalidK}}).TTP_{5a} \\
& \quad + (\text{timeoutTTP}, r_{\text{timeoutTTP}}).TTP_6 \\
& \quad + (\text{sendTTPreason}, r_{\text{sendTTPreason}}).TTP_{4a} \\
&TTP_{4a} \overset{\text{def}}{=} (\text{sendTTPvalidK}, r_{\text{sendTTPvalidK}}).TTP_5 \\
&TTP_5 \overset{\text{def}}{=} (\text{sendMpkeyTTP}, r_{\text{sendMpkeyTTP}}).TTP_6 \\
&TTP_{5a} \overset{\text{def}}{=} (\text{forwardKtoC}, r_{\text{forwardKtoC}}).TTP_0 \\
&TTP_6 \overset{\text{def}}{=} (\text{sendCkByTTP}, r_{\text{sendCkByTTP}}).TTP_7 \\
&TTP_7 \overset{\text{def}}{=} (\text{investigationIvalidCoinToB}, r_{\text{investigationIvalidCoinToB}}).TTP_8 \\
&TTP_8 \overset{\text{def}}{=} (\text{takeActionAgainstM}, r_{\text{takeActionAgainstM}}).TTP_0 \\
&TTP_9 \overset{\text{def}}{=} (\text{investigationIvalidCoinToB}, r_{\text{investigationIvalidCoinToB}}).TTP_10 \\
&TTP_{10} \overset{\text{def}}{=} (\text{investigationIvalidCoinToB}, r_{\text{investigationIvalidCoinToB}}).TTP_3 \\
&TTP_{10} \overset{\text{def}}{=} (\text{investigationIvalidCoinToB}, r_{\text{investigationIvalidCoinToB}}).TTP_9 \\
&TTP_{10} \overset{\text{def}}{=} (\text{investigationIvalidCoinToB}, r_{\text{investigationIvalidCoinToB}}).TTP_0 \\
\end{align*}
\]

The above model component specifies TTC's different behaviours. TTP has thirteen states. TTP moves from states \(TTP_0\) to \(TTP_{10}\) to solve the dispute between C and M. The actions are preformed based on the specified rates in order for TTP to involve in the interaction and provide a fair resolution for the disputed parties. TTP's main actions are \(\text{download}, \text{validateCoinToB}, \text{timeoutTTP}, \text{sendMpkeyTTP}, \text{sendCkByTTP}, \text{investigationIvalidCoinToB}, \text{takeActionAgainstM}\) and \(\text{discoverMisbehavingC}\). TTP controls the rates of those actions.

Bank component

\[
\begin{align*}
&B \overset{\text{def}}{=} (\text{requestBDigitalCoins}, r_{\text{requestBDigitalCoins}}).B \\
& \quad + (\text{sendCDigitalCoins}, r_{\text{sendCDigitalCoins}}).B + (\text{sendBCoinByM}, r_{\text{sendBCoinByM}}).B \\
& \quad + (\text{sendMyes}, r_{\text{sendMyes}}).B + (\text{sendMno}, r_{\text{sendMno}}).B \\
& \quad + (\text{cspentTheCoinToTTP}, r_{\text{cspentTheCoinToTTP}}).B \\
& \quad + (\text{mspentTheCoinToTTP}, r_{\text{mspentTheCoinToTTP}}).B + (\text{validateCoinToB}, r_{\text{ve}}).B \\
& \quad + (\text{sendTTPyes}, r_{\text{yes}}).B + (\text{sendTTPno}, r_{\text{no}}).B \\
& \quad + (\text{investigationIvalidCoinToB}, r_{\text{investigationIvalidCoinToB}}).B \end{align*}
\]

The last part of the model is for B component. B just has one state. The B’s main actions to support the purchase processes between the components C and M are \(\text{sendCDigitalCoins}, \text{sendMyes}\) and \(\text{sendMno}\) and to support the dispute resolution are \(\text{sendTTPyes}, \text{sendTTPno}, \text{cspentTheCoinToTTP}\) and \(\text{mspentTheCoinToTTP}\) as described in the scenarios of the extended protocol specification (Subsection 2.2). The rates of these actions are controlled by B.
The system equation  The system equation and complete specification are given by

\[
System \triangleq TTP[K] \bigotimes_R \left( B[S] \bigotimes_M \left( C_0[N] \bigotimes_L M_0[N] \right) \right)
\]

Where the cooperation sets \( R = \{\text{download, sendTTPinfo, validateCoinToB, sendTTPyes, sendTTPno, askMforValidK, sendTTPvalidK, timeoutTTP, sendTTPreason, sendMpkbyTTP, forwardKtoC, sendCkByTTP, investigationInvalidCoinToB, takeActionAgainstM, cspentTheCoinToTTP, mspentTheCoinToTTP, discoverMisbehavingC} \} \), \( M = \{\text{requestBDigitalCoins, sendCDigitalCoins, sendMno, sendMyes} \} \) and \( L = \{\text{sendMPO, sendCEP, sendCAbort, sendMCoinDk, sendMACbort, sendCPDk, complete, cTimeoutExpired} \} \), any action in the lists \( R, L \) and \( M \) is shared action between the components specified in the system equation. \( N \) is the number of clients and merchant copies on the system, \( K \) is the number of TTPs, \( S \) is the number of Bs. The four components are initially in the states \( TTP_0, C_0, M_0 \) and \( B \).

Furthermore, the rates of all the main actions carried out by \( M \) depend on the number of Cs interacting with \( M \). The rates of \( M \)'s main activities are divided by the number of Cs that interact with it. The \( M \)'s main actions are \( \text{sendCEP, sendCAbort, startContactB, sendBCoinByM, sendCPDk, complete, cTimeoutExpired} \). For example, the rate of \( \text{sendCEP} \) action is calculated as follows:

\[
r_{\text{sendCEP}} = \frac{r_{\text{sendCEP}1}}{N}
\]

Also, the service rates of all the main actions of \( B \) are calculated based on the number of \( C \) and \( M \)'s copies as well as the number of Bs involved in the interaction. The \( B \)'s main actions are \( \text{sendCDigitalCoins, sendMno, sendMyes, mspentTheCoinToTTP, cspentTheCoinToTTP, sendTTPyes and sendTTPno} \). One, two or more Bs can be involved in the protocol to serve \( C \) and \( M \). For example, the rate of \( \text{sendCDigitalCoins} \) action is calculated as follows:

\[
r_{\text{sendCDC}} = \left( \frac{r_{\text{sendCDigitalCoins}1}}{N} \right) \ast S
\]

Further, the service rates of all TTP’s main actions depend on the number of both Cs and TTPs interacting with each other. TTP’s main actions are \( \text{forwardKtoC, download, sendCkByTTP, askMforValidK, takeActionAgainstM, discoverIncorrectPTK, sendMpkbyTTP, validateCoinToB, sendMpkbyTTP} \) and \( \text{investigationInvalidCoinToB} \). One, two or more TTPs can be involved in the protocol [6]. For example, the rate of is \( \text{download} \) action calculated as follows:

\[
r_d = \left( \frac{r_{\text{download}}}{N} \right) \ast K
\]

4 Performance evaluation of the extended protocol

We seek to calculate the average response times of TTPs when serve \( C \) to solve the dispute. We assigned 1 as a value for all rates. The main actions of \( M \) are
calculated based on the number of customers in the system and the main actions of TTP are calculated based on the number of customers and TTP in the system, as mentioned in Section 3.

In Sub-figure 2a, the number of TTP involved in the system is 20 and we change the number of customers from 200 to 1000 to show how increasing the number of the customer seeking help from TTP would impact the performance of the protocol. The average response time of TTP for all main states ($TP_{5a}$, $TP_{6}$ and $TP_{10}$) to solve the dispute is the same. Having a large number of customers significantly increases the average response time of TTPs which creates more performance overhead. However, in Sub-figure 2b, the number of TTP is increased to 40 which cause a decrease in the response time in relation to the number of customers in the system compared to Sub-figure 2a. Therefore, having a larger number of TTPs involved in the protocol when the dispute occurs between C and M mitigates the security protocol’s performance overhead.

We are also interested in investigating the population level analysis of $C_{4}$, $C_{6}$, $C_{9}$ and $C_{10}$ ($C_{4}$ and $C_{6}$ for having a service from M and $C_{9}$ and $C_{10}$ for interacting and having a service from TTP) and throughput analysis of some main actions that provide service to C. Both analyses are studied in relation to different probabilities of M to be honest and different population numbers of Cs and M’s copies.

In Sub-figures 3a and 4a, the probabilities for M to be honest are changed from $p = 0.1$ to $p = 0.9$, the number of TTP is 20, the number of C and M’s copies is 200 and the number of banks is 20. Sub-figure 3a shows how decreasing the probabilities of M to be honest has a clear effect on increasing the number of $C_{10}$ copies. So more customers seek dispute resolution. However, the number of C in $C_{4}$ and $C_{6}$ does not experience any change but there are more Cs in $C_{6}$ waiting to get a decryption key for the encrypted product than in $C_{4}$ to get an encrypted product. Sub-figure 4a illustrates the significant increase in the throughput of the TTP’s actions that provide the dispute resolution when the

Fig. 2: The average response time of $TP_{5a}$, $TP_{6}$ and $TP_{10}$ using ODE when S=20 and the probability of M to be honest is 0.1.
probabilities of M to be honest is decreasing. We believe this is because more customers are seeking help from TTP.

Fig. 3: The population level analysis using ODE with K=20 and S=20 in relation to different probabilities of M to be honest.

Fig. 4: The throughput analysis of actions using ODE with K=20 and S=20 in relation to different probabilities of M to be honest.

In Sub-figures 3b and 4b, the number of C and M's copies (N) is increased to 600. Increasing the number of Cs in a system has a clear impact on the population level and the actions' throughputs. Just like Sub-figure 3a, Sub-figure 3b shows how decreasing the probabilities of M to be honest has a clear effect on increasing the average number of $C_{10}$ as more customers seek a dispute resolution. Also, the number of Cs in $C_4$ and $C_6$ does not experience any change, but more Cs in $C_6$ waiting to get a decryption key for the encrypted product than in $C_4$ to get an encrypted product. However, increasing the number of Cs in the system does not significantly impact $C_9$ and $C_{10}$. The impact is clearly on $C_4$ and $C_6$, compared to $C_4$ and $C_6$ in Sub-figure 3a.
Moreover, Sub-figure 4b shows the increase in the throughput of the TTP’s actions that provide the dispute resolution when the probabilities of M to be honest is decreasing. We believe this is because more customers are seeking help from TTP. Moreover, all actions have a clear reduction on their throughput when we increased the number of C. The throughput of the TTP’s actions are less than the throughput of the TTP’s actions in Sub-figure 4a when the the number of Cs in a system is 200. Therefore, larger number of customers in the system will have an affect on the TTP’s responses.

Figure 5 shows how faster the system settled in relation to the population number of Cs and M’s copies and the probabilities of M to be honest. The larger the population number is and the less probability of M to be honest is, the longer time will be taken for the system to settle.

![Figure 5: The steady-state detection time in relation to the population number (200 and 600).](image)

Now we interested in changing the rates of the shared actions between M and B. The shared actions between M and B are $rsendBCoinByM$, $rsendMyes$ and $rsendMno$. We increased and decreased the rates to show how these would have an impact on the system performance. First, the shared actions rates between M and B are decreased to $r=0.2$ and $r=0.5$. Then they are increased to $r=2$ and $r=4$. The rates are calculated depend on the number of N and S involved in the system, as follows:

$$rsendBCoinByM = r/N$$
$$rsendMyes = (r/N) \times S$$
$$rsendMno = (r/N) \times S$$

Where N is the number of Cs and M’s copies and S is the number of banks.

The following figure shows the population level analysis of $C_4$ and $C_6$ for having a service from M and $C_9$ and $C_{10}$ for interacting and having a service from TTP. The probability for M to be honest is changed from $p = 0.1$ to $p = 0.9$. 

Sub-figures 6a and 6b illustrate that when the shared actions rates between M and B to check the C's digital coin’s validity are slow, C experiences a big delay in receiving a product decryption key. There are large waiting Cs in C_6. In Sub-figures 6a and 6b, you can notice that the population levels of C_9 and C_10 are slightly decreased. We believe this is because more Cs waiting in C_4 and C_6 to be served before moving to C_9 and C_10. So the faster the rate, the less delay would be, as shown in Sub-figures 6c and 6d. Moreover, Sub-figures 6c and 6d show a significant increase in the population levels of C_10.

Fig. 6: The population level analysis using ODE with K=20, N=200 and S=20 and with different rates for the shared actions between M and B.

Figure 7 shows the throughput of some main actions to serve a customer. These actions are discoverMisbehavingC, forwardKtoC, sendCkByTT and sendTTinfo to interact and seek a help form TTP and sendCAbort, sendCEP and sendCPDk to get a service form M and cTimeoutExpired. In Figure 7, the probabilities for M to be honest are changed from p = 0.1 to p = 0.9, the number of TTP is 20, the number of C and M’s copies is 200 and the number of banks is 20. The shared actions rates between M and B are decreased to r = 0.2 and r = 0.5 and then increased to r = 2 and r = 4. The rates are calculated based on the number of N and S involved in the system, as mentioned in this Section.
Figure 7 illustrates a significant increase in the throughput of the TTP’s actions that provide the dispute resolution when M’s probabilities to be honest are decreasing in all Sub figures. The TTP’s actions are `discoverMisbehavingC`, `forwardKtoC`, `sendCkByTTP` and `sendTTPinfo`. We believe this is because more customers are seeking help from TTP when M’s probabilities to be honest are lower. Moreover, there is a considerable improvement in all actions’ throughputs when the shared actions rates between M and B to check the C’s digital coin’s validity are higher.

![Throughput Analysis](image)

Fig. 7: The throughput analysis using ODE with K=20, N=200 and S=20 and with different rates for the shared actions between M and B.

5 Conclusion

This investigation explores an anonymous and failure resilient fair-exchange e-commerce protocol when the misbehaviour and connections problem occurs. The involvement of TTP becomes essential to resolving disputes between participants. We propose the PEPA model for the protocol. The evaluation results indicated that when the protocol preserved customer anonymity, this introduced extra performance cost. Furthermore, when there is a dispute between participants, the TTP involvement is active, and this would introduce extra load on TTP and/or Bs, which would influence the system performance. Moreover, we
showed how scaling up TTP and B resources to handle escalating misbehaving party mitigates the negative impact on the system performance. In this paper, we concentrate on the misbehaviour of the merchant. In our future work, we will consider models and scenarios when misbehaviour occurs from customers. Also, we will consider PEPA models of malicious misbehaviour where an adversary’s behaviour changes over time and the system needs to respond in kind in order to remain secure and to provide a sustainable level of performance to the legitimate users by, for example, scaling of the resources to handle escalating threats without a negative impact on the rest of the system.

References