Performance Modelling of an Anonymous and Failure Resilient Fair-Exchange E-Commerce Protocol

Ohud Almutairi
School of Computer Science
Newcastle University
United Kingdom
o.m.m.almutairi2@newcastle.ac.uk

Nigel Thomas
School of Computer Science
Newcastle University
United Kingdom
nigel.thomas@newcastle.ac.uk

ABSTRACT

This paper explores a type of non-repudiation protocol, called an anonymous and failure resilient fair-exchange e-commerce protocol, which guarantees a fair-exchange between two parties in an e-commerce environment. Models are formulated using the PEPA formalism to investigate the performance overheads introduced by the security properties and behaviour of the protocol. The PEPA eclipse plug-in is used to support the creation of the PEPA models for the security protocol and the automatic calculation of the performance measures identified for the protocol models.

KEYWORDS

Secure e-commerce; PEPA; performance modelling

ACM Reference format:

Ohud Almutairi and Nigel Thomas. 2019. Performance Modelling of an Anonymous and Failure Resilient Fair-Exchange E-Commerce Protocol. In *Proceedings of Tenth ACM/SPEC International Conference on Performance Engineering (ICPE '19), April 7-11, 2019, Mumbai, India. ACM,* New York, NY, USA, 8 pages. https://doi.org/10.1145/3297663.3310310

1. Introduction

The need for a secure system leads to the development of many different security protocols and algorithms. However, security mechanisms impose additional computational costs on a system, possibly adversely affecting performance. Modelling and measuring the performance of security algorithms and protocols can help to develop a system with acceptable levels of both security and performance and support the development of lightweight protocols.

However, the steps taken to make systems more secure can affect their overall performance. Security 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 [1], and the computational cost of security mechanisms adds performance costs to wireless sensor networks during secure communication [2].

Improved system performance can be achieved by lowering the computational cost of security functions, i.e. decreasing the length of the key used in encryption and decryption can reduce the computation cost and subsequently increase the performance of the system [3]. However, this approach can influence the security status of the system. Thus, it is important to develop a system affording an optimal balance between security and performance concerns.

Performance and security are essential aspects for almost all systems. The methods used in order to explore and measure those aspects are either experimental, as in [4] or model-based, employing modelling techniques, such as Stochastic Petri Nets (SPN) [5], Stochastic Process Algebra (SPA) [6] and Queueing Theory [7]. Moreover, employing a model-based approach could prove more flexible and beneficial for evaluating performance and security aspects of the system, as it replaces the pre-existing system with one that can be readily understood, studied and analysed, making it easier to complete modifications to investigate the different results that are possible [8].

The approach used to model a protocol under investigation in this paper is Performance Evaluation Process Algebra (PEPA). PEPA is a well-known implementation of SPA. A system is modelled in PEPA as a set of components which interact and engage individually or with other components in activities to evaluate its performance [9]. Thus, the components represent the active parts in the system and the behaviour of each part is represented by its activities.

This investigation considers a type of e-commerce protocol called an anonymous and failure resilient fair-exchange e-commerce protocol [10], which is a non-repudiation security protocol implemented during e-commerce transactions. In an electronic commerce environment, two or more parties interact with each other to exchange products. This type of security protocols has been developed to ensure fair exchange between participants and that no party can take advantage over the other party during the exchange process [11]. Understanding the behaviour of these protocols and the performance cost they introduce could enable the development of lightweight e-commerce protocols. The aim of this paper is to build models of the anonymous and failure resilient fair-exchange e-commerce

ICPE '19, April 7–11, 2019, Mumbai, India

ACM ISBN 978-1-4503-6239-9/19/04…$15.00
https://doi.org/10.1145/3297663.3310310

protocol which is proposed by Ray *et al* [10] in order to understand its behaviour, evaluate the performance cost it introduces and understand how adding more security features, such as anonymity, will introduce an extra performance overhead.

The creation and analysis of PEPA models are supported by the PEPA Eclipse plug-in [12]. This tool has been developed to support Markovian steady-state analysis, stochastic simulation, and Ordinary Differential Equations (ODE) analysis of PEPA models in the Eclipse Platform [12].

The paper is organized as follows. Section 2 presents some existing related studies that have reviewed the interrelation between performance and security. In Section 3, different descriptions and PEPA models of the protocol under our study are introduced and each proposed model is followed by its evaluation and results. Finally, Section 4 concludes the report by providing an overview of the study findings and future work.

2. Related Works

There are a number of researchers have studied and measured the performance of security related algorithms and protocols. For example, in [4], Wolter and Reinecke studied the interrelation of security with performance, examining how increasing one affects the other in a model-based evaluation using Generalized Stochastic Petri Net (GSPN) formalism and the TimeNET tool. The issue their model explored was the impact of different encryption key lengths on system performance. They were the first to build a simple general model to explore the trade-off between performance and security. However, their model was formulated as a simple general framework and so does not accurately represent a complex system.

As in the previous study, Cho *et al* [5] have proposed a system model based on SPN formalism. They outline a mathematical model for a secure group communication system in mobile ad hoc networks (MANETs) to analyse performance-security trade-off. The MANETs environment suffers high security vulnerability as it is an open medium [5]. Their results clearly show how best to provide an optimal setting of security techniques in their system.

Montecchi *et al* [13] studied the trade-off between scalability, performance and the security aspects of a multi-service web-based platform. Stochastic Activity Networks (SANs) is an extension of SPN, was adopted to construct a model of the target system. However, Montecchi *et al* mainly focus on studying and exploring the scalability of a system with respect to performance, and the security aspect was considered as having an indirect effect on scalability, through its influence on performance. They show that employing the chosen security mechanisms to the studied system would have a significant influence on system performance, which in turn would be expected to influence system scalability.

The security impact on system performance has also been studied using PEPA. This formalism is employed by Zhao and Thomas in [14] to model two security protocols. Zhao and Thomas model Zhou and Gollman's non-repudiation protocols which are a type of security protocol, designed to ensure a fair exchange between two or more parties. They clearly illustrate how performance could be influenced by a different number of clients requesting a service from a Trust Third Party (TTP) in the protocols.

Elsewhere, Zhao and Thomas [6] conducted a performance study of another type of non-repudiation protocol, called an optimistic fair exchange protocol. This type of security protocol employs a TTP, when an unfair exchange occurs in an e-commerce environment. They proposed PEPA models to investigate the performance costs introduced by the protocol. They clearly show that a misbehaviour event leading to TTP involvement increases the performance cost of a system compared to cases in which there is no misbehaviour between the participants.

Furthermore, the performance of encryption algorithms was studied by Lamprecht *et al* in [4]. Unlike the previously outlined studies, their study implemented the algorithms in Java, rather than proposing a specific system model. They conducted a comparative performance evaluation of common encryption algorithms common to online transactions. They measured operation time as the performance metric when testing each of the algorithms; i.e. key generation, encryption and decryption. The findings of their study can aid to select a suitable encryption algorithm for an online transaction system and associated applications. However, as their study is not model-based, they also indicate how different implementations for the same algorithm could result in different runtimes. Therefore, this might not be accurate enough to generalize from.

Finally, although many studies have been conducted, more research is needed regarding modelling and investigating the performance cost introduced by security protocols in order to support the development of security protocols that offer acceptable levels of both security and performance.

3. An Anonymous and Failure Resilient Fair-Exchange Protocol

An anonymous and failure resilient fair-exchange e-commerce protocol was proposed by Ray *et al* [10]. This protocol guarantees a fair-exchange between two parties. It satisfies the following features: first, fairness – no party can have any advantages over the other party during the exchange course; second, anonymity–the parties, a customer and/or a merchant can interact without disclosing any personal information; third, no manual dispute resolution; fourth, not relying on the service of a single trust 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.

The protocol relies on TTPs but does not need them to be active at any time except if a problem occurs. With offline TTP, 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 the resolution of the problem and ensure fair-exchange. Following the description provided by Ray *et al* [10], this paper first presents a PEPA model of a failure

resilient fair-exchange protocol without a customer anonymity feature. Then, a PEPA model of the protocol with a customer anonymity feature is presented and evaluated. The first and second protocol are modelled with no dispute between parties. In addition, the discussion focuses on the behaviour aspects of the protocols to analyse their performance.

3.1 Failure resilient fair-exchange protocol

This protocol preserves all the features identified above except the customer's anonymity; the true identity of a party can be disclosed by obtaining a payment token, as described below.

3.1.1 Basic failure resilient fair-exchange protocol specification

A formal description of this protocol with the security related details is given in [10]. The basic protocol with no misbehaviour of any parties is as follows:

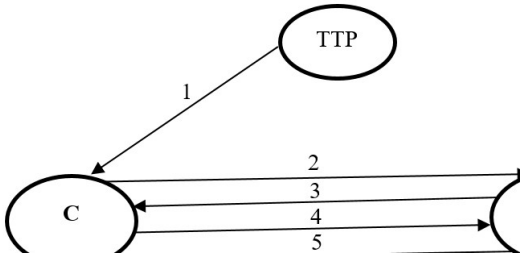

Figure 1: The basic protocol.

Before the protocol starts, the environment need to be set up with the following two steps:

- A customer (C) needs to create an account with a Bank (B). B creates a pair of keys. One is sent to C and the other is kept by B.
- A merchant (M) needs to register with a Trust Third Party (TTP). TTP creates a pair of keys. One is sent to M and the other is kept by TTP. For every products M wants to sell, M sends the products and its description to TTP. TTP then encrypts the electronic product with the same key that it sent to M before uploading the product to its website to be advertised.

Then, the protocol has the following five interaction steps, the words in bold are the actions name that we used in our PEPA model:

1. ***download*** (TTP->C): C visits the TTP website and downloads the encrypted electronic product from the TTP server. C cannot obtain the product without a decryption key. This encrypted product can be used to validate the product received from M. Then C contacts M and begins an interaction with them by sending the next message.
2. ***sendMPO*** (C->M): C sends a message containing the purchase order (PO) to M. This message contains the payment token (PT), and the identity of its Bank (B).
3. ***sendCEP*** or ***sendCAbort*** (M->C): M sends the encrypted product to C or sends a transaction abort statement. After M receives the message from C containing the PO (step 2), it checks it. If M is satisfied, it sends the encrypted product to C.
4. ***sendMPTDk*** or ***sendMAbort*** (C->M): C sends the decryption key for the PT or a abort statement to M. After C receives the message from M (step 3), C checks it. If it contains an abort statement, then C aborts the transaction. If it contains the encrypted product, then C validates it with the encrypted product received from the TTP (step 1). If the product is valid, C sends the decryption key for the PT, encrypted with M's public key and sets a timer. If the product is not valid, C sends an abort to M.
5. ***sendCPDk*** (M->C): M sends the product decryption key to C or ends the transaction. On receiving the message from C, M checks it. If the message contains an abort, then M terminates the transaction. If the message contains the decryption key for the PT, M obtains the PT and then sends the decryption key for the product to C. The decryption key is encrypted with C's public key.

3.1.2 A PEPA model of the basic failure resilient fair-exchange e-commerce protocol

In our PEPA model, there are three types of components: customer (C), merchant (M) and trust third party (TTP). The model comprises of 3 parts, one for each component. C and M move sequentially from their different behaviours based on the activities specified in the model. The model is formulated as follows:

$$M_0 \stackrel{\text{def}}{=} (\text{sendMPO}, r_{sendMPO}).M_1$$
$$M_1 \stackrel{\text{def}}{=} (\text{sendCEP}, r_{\text{sendCEP}}).M_2 + (\text{sendCAbort}, r_{\text{sendCAbort}}).M_5$$
$$M_2 \stackrel{\text{def}}{=} (\text{sendMPTdk}, r_{sendMPTdk}).M_3 + (\text{sendMAbort}, r_{sendMAbort}).M_4$$
$$M_3 \stackrel{\text{def}}{=} (\text{sendCPDk}, r_{\text{sendCPDk}}).M_4$$
$$M_4 \stackrel{\text{def}}{=} (\text{complete}, r_{complete}).M_0$$
$$M_5 \stackrel{\text{def}}{=} (\text{sendMAbort}, r_{sendMAbort}).M_4$$

Above model component specifies M's different behaviours, moving from M0 to M5. When M is in state M0 (step 2 in protocol's description), M performs action *sendMPO* at rate $r_{sendMPO}$ leading to M1. Then, in state M1 (step 3 in protocol's description), either action *sendCEP* at rate r_{sendCEP}, leading to M2 could happen or action *sendCAbort* at rate $r_{\text{sendCAbort}}$ leading to M5. In state M2 (step 4), M can perform either action *sendMPTdk* at rate $r_{sendMPTdk}$ leading to M3 or action *sendMAbort* at rate $r_{sendMAbort}$ leading to M4. Then, in state M3 (step 5), the only action happens is *sendCPDk* at rate r_{sendCPDk} leading to M4 which is the state when M performs action *complete* at rate $r_{complete}$ leading back to M0 which it means that the exchange between C and M has finished. The state M5 is when M performs action *sendMAbort* as a result of performing action *sendCAbort* in state M1 leading to M4. This is to keep smooth communication between M and C. Therefore when M sends abort to C (in M1), C also sends abort to M (in M5) and then moves to the last state which is M4 to terminate the interaction. After M4 the behaviour returns to M0 so that the model becomes cyclic and steady state measures can be obtained.

$C_0 \stackrel{\text{def}}{=} (download, r_d).C_1$
$C_1 \stackrel{\text{def}}{=} (\text{sendMPO}, r_{sendMPO}).C_2$
$C_2 \stackrel{\text{def}}{=} (\text{sendCEP}, r_{\text{sendCEP}}).C_3 + (\text{sendCAbort}, r_{\text{sendCAbort}}).C_6$
$C_3 \stackrel{\text{def}}{=} (\text{sendMPTdk}, r_{sendMPTdk}).C_4 + (\text{sendMAbort}, r_{sendMAbort}).C_5$
$C_4 \stackrel{\text{def}}{=} (\text{sendCPDk}, r_{\text{sendCPDk}}).C_5$
$C_5 \stackrel{\text{def}}{=} (\text{complete}, r_{complete}).C_0$
$C_6 \stackrel{\text{def}}{=} (\text{sendMAbort}, r_{sendMAbort}).C_5$

Above component represents C's different behaviours, moving from C0 to C6. First state is C0 (step 1 in protocol's description). It is the state when C visits TTP website and performs action *download* for a specific product at rate r_d leading to C1. Then, in state C1 (step 2 in protocol's description), the only action happens is *sendMPO* at rate $r_{sendMPO}$ leading to C2. In state C2 (step 3), one of two action could happen, either *sendCEP* at rate r_{sendCEP} leading to C3 or *sendCAbort* at rate $r_{\text{sendCAbort}}$ leading to C6. In state C3 (reflects step 4), also there is one of two action could happen, either *sendMPTdk* at rate $r_{sendMPTdk}$ leading to C4 or *sendMAbort* at rate $r_{sendMAbort}$ leading to C5. In state C4 (step 5 in protocol's description), the only *sendCPDk* can occur at rate r_{sendCPDk} leading to C5. The state C5 is a termination step, when C performs action *complete* at rate $r_{complete}$ leading back to C0 to be able to finish the interaction and maybe use the product before starting again. In state C6, C performs action *sendMAbort* at rate $r_{sendMAbort}$ as a result of performing action *sendCAbort* (receiving abort from M). Performing C6's action leads to C5. This step allows C to send a response to M and starts the interaction again providing a correct information.

$TTP \stackrel{\text{def}}{=} (download, r_d).TTP$

In the model, TTP has one state. In state TTP (step 1 in protocol's description), the only action could happen is *download* at rate r_d leading to the same state TTP.

The system equation and complete specification are given by

$\text{System} \stackrel{\text{def}}{=} TTP[K] \bowtie_J C_0[N] \bowtie_L M_0[N]$

Where J={download}, L={sendMPO, sendCEP, sendCAbort, sendMPTdk, sendMAbort, sendCPDk, complete}, any action in list J and L is a shared action between the components specified in system equation. N is the number of C and M instances in the system, K is the number of TTPs. The three components are initially in the states TTP, C0 and M0.

M has multiple copies, each copy is associated with one C in order to serve it. This indicates that the rates of the main activities conducted by M are divided by the number of Cs that interact with it. M activity rates are calculated as follows:

$$r_{\text{sendCEP}} = \frac{r_{\text{sendCEP1}}}{N}$$
$$r_{\text{sendCAbort}} = \frac{r_{\text{sendCAbort1}}}{N}$$
$$r_{\text{sendCPDk}} = \frac{r_{\text{sendCPDk1}}}{N}$$

3.1.3 *Performance evaluation of the basic protocol*

The current investigation seeks to calculate the average response time of the merchant that customers will observe. The rate of all actions is 1.

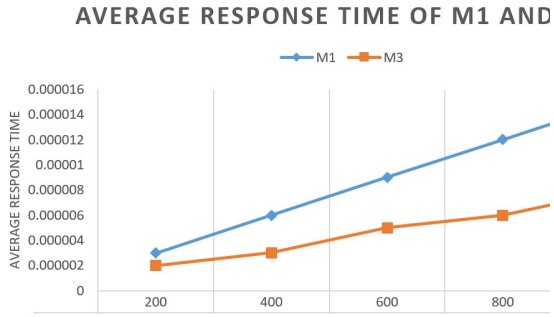

Figure 2: Average response time of M1 and M3 using ODE.

Figure 2 shows how increasing the number of customers affected the merchant's average response time for actions *sendCEP*, *sendCAbort*, and *sendCPDk*. M1 and M3 are the states of merchant when performs those actions to response to customer during the interaction course.

Furthermore, the population level analysis that shows the average number of C's copies in state C2 and C4 for requesting a service from M is shown in Figures 3 and 4:

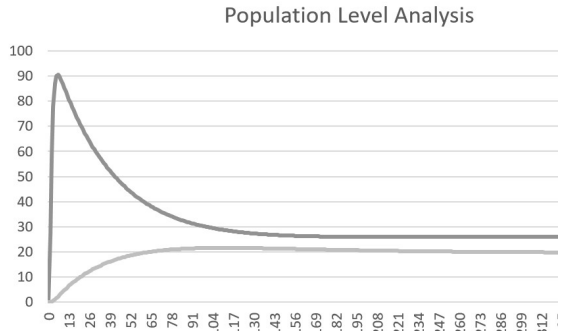

Figure 3: The population level analysis using ODE with *K*=100 and *N*=100.

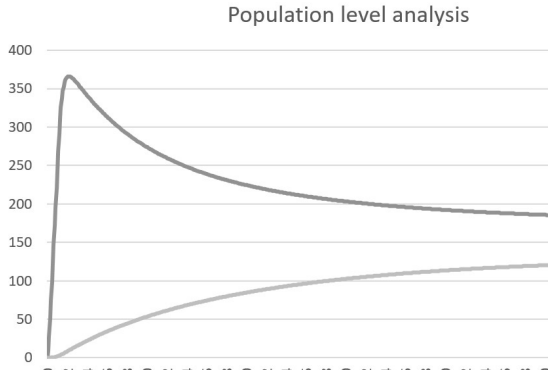

Figure 4: The population level analysis using ODE with *K*=100 and *N*=400.

The average number of copies of C2 increased when *N* (the number of C and M) was increased (Figures 3 and 4). Moreover, more customers waiting to be served in C2 than in C4. This means that some customers abort the process and would not reach the state C4.

3.2 Optimistic anonymous protocol

This version of the protocol ensures that customer privacy is protected from any other parties. The customer does not need to share any personal information with a merchant in order to buy. Thus, the customer's true identity is hidden from the merchant. In the protocol described previously (see Section 3.1), the payment token that the customer sends contains some personal information, such as the identity and bank account information. Therefore, the merchant will have the detailed personal information of the customer once they receive the payment token. This will deter some customers from buying from some merchants as they are not willing to share these personal details. Thus, as well as delivering all the features provided by the failure resilient fair-exchange protocol (Section 3.1), the optimistic anonymous protocol also preserves the customer's anonymity.

Ray *et al* [10] 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 [15]. The customer uses digital base money to buy from merchants. By using this method, merchants are not able to obtain any personal information from the customer or create a customer profile without a permission.

3.2.1 Optimistic anonymous protocol specification

The formal description of the protocol and security related details are provided in [10]. The following is an informal description of the protocol. As with the basic protocol, before the protocol is initiated, the environment need to be set up with the same initial steps detailed in Section 3.1.1. Unlike the basic failure resilient fair-exchange protocol, 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 optimistic anonymous protocol has the following nine interaction steps [10]:

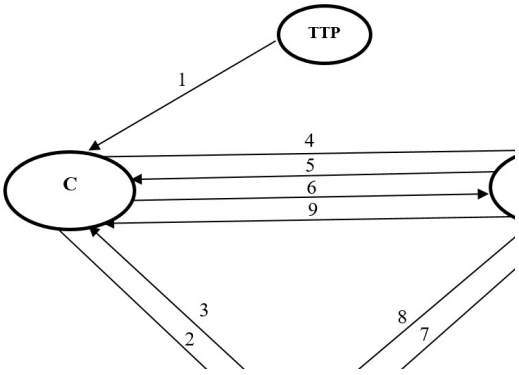

Figure 5: The optimistic anonymous protocol.

1. ***download*** (TTP->C): same as step 1 in the protocol description (Section 3.1.1). However, if C is interested in the product, they must contact their bank (B) to request digital coins to buy the product with, as per the following message.
2. ***requestBDigitalCoins*** (C-> B): C sends a request to B for digital coins. C's request message contains an unsigned blinded coin, the true identity of the customer, and the account number.
3. ***sendCDigitalCoins*** (B-> C): Once B has received the request message, C's bank account is debited for the same amount of money as the value of the unsigned blinded coin. Then B generates the digital coin by signing the blinded coin, and sends the digital coin to C.
4. ***sendMPO*** (C`-> M): C sends a message containing the purchase order (PO) to M. The PO contains the customer's pseudo identifier (C`) and the product price. This message also contains the digital coin encrypted with C's secret key. Note that the customer uses a one-time private/public key pair.
5. ***sendCEP*** or ***sendCAbort*** (M-> C`): M sends the encrypted product to C` or sends an abort statement to end the transaction after receiving the message from C` containing the PO (*step 4*). This step is similar to step 3 in the basic protocol description (Section 3.1.1).
6. ***sendMCoinDk*** or ***sendMAbort*** (C`-> M): C` sends the decryption key of the digital coin to M or sends an abort to end the transaction. After receiving the message from M (*step 5*), C` checks it. If it contains an abort or invalid encryption product, then C` aborts the transaction. If it contains the encrypted product, then C` validates it with the encrypted product received from TTP (*step 1*). If the product is valid, C` sends the decryption key for the digital coin, which has been encrypted with M's public key, 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 invalid, C` sends an abort statement to M.
7. ***sendBCoinByM*** or ***sendCAbort*** (M->B or M->C`): M sends B their identity and the signed digital coin, or M sends C` an abort to terminate the transaction. On receiving the decryption key for the digital coin from C`, M checks B's signature on the coin and whether or not the amount received is equal to the price of the product. If M is satisfied, M sends B their identity and the signed digital coin. 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'. Once B receives the coin from M, B checks whether or not the coin has been spent. 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 product decryption key to C` after receiving 'yes' from B, or ends the transaction by sending an abort to C` after receiving 'no' from B. On receiving the 'yes' from B, M sends the product decryption key which is encrypted with C`'s public key.

3.2.2 PEPA models of the optimistic anonymous protocol

A PEPA model of the optimistic anonymous protocol model contains 4 components which are merchant (M), customer (C), trust third party (TTP) and bank (B). The model is formulated as follows:

$M_0 \stackrel{\text{def}}{=} (sendMPO, r_{sendMPO}).M_1$
$M_1 \stackrel{\text{def}}{=} (sendCEP, r_{sendCEP}).M_2$
$\quad + (sendCAbort, r_{sendCAbort}).M_6$
$M_2 \stackrel{\text{def}}{=} (sendMCoinDk, r_{sendMCDk}).M_3$
$\quad + (sendMAbort, r_{sendMAbort}).M_6$
$M_3 \stackrel{\text{def}}{=} (sendBCoinByM, r_{sendBCByM}).M_4$
$\quad + (sendCAbort, r_{sendCAbort}).M_8$
$M_4 \stackrel{\text{def}}{=} (sendMyes, r_{sendMy}).M_5 + (sendMno, r_{sendMn}).M_7$
$M_5 \stackrel{\text{def}}{=} (sendCPDk, r_{sendCPDk}).M_6$
$M_6 \stackrel{\text{def}}{=} (complete, r_{complete}).M_0$
$M_7 \stackrel{\text{def}}{=} (sendCAbort, r_{sendCAbort}).M_8$
$M_8 \stackrel{\text{def}}{=} (sendMAbort, r_{sendMAbort}).M_6$

Above part of the model is for M component. The first two states are same as states M0 and M1 described in the basic protocols model. The states M0 and M1 reflect step 4 and step 5 of the optimistic anonymous protocol description, respectively. In state M2, one of two actions could happen either *sendMCoinDk* at rate $r_{sendMCDk}$ leading to M3 if customer receives an encrypted product or *sendMAbort* at rate $r_{sendMAbort}$ leading to M6 as a result of performing *sendCAbort* in M1. This state reflects step 6 of the optimistic anonymous protocol description. When M reaches state M3, one of two actions could happen either *sendBCoinByM* at rate $r_{sendBCByM}$ leading to M4 if M is satisfied with the amount of the digital coins or *sendCAbort* at rate $r_{sendCAbort}$ leading to M8 if M is not satisfied, as described in step 7. In state M4, either actions could be performed *sendMyes* at rate r_{sendMy} leading to M5 to have a confirmation from B that the coins is valid or *sendMno* at rate r_{sendMn} leading to M7 in order to send an abort to customer when M has a confirmation from bank that the coins is invalid, as described in step 8.

$C_0 \stackrel{\text{def}}{=} (download, r_d).C_1$
$C_1 \stackrel{\text{def}}{=} (requestBDigitalCoins, r_{requestBDC}).C_2$
$C_2 \stackrel{\text{def}}{=} (sendCDigitalCoins, r_{sendCDC}).C_3$
$C_3 \stackrel{\text{def}}{=} (sendMPO, r_{sendMPO}).C_4$
$C_4 \stackrel{\text{def}}{=} (sendCEP, r_{sendCEP}).C_5 + (sendCAbort, r_{sendCAbort}).C_8$
$C_5 \stackrel{\text{def}}{=} (sendMCoinDk, r_{sendMCDk}).C_6$
$\quad + (sendMAbort, r_{sendMAbort}).C_7$
$C_6 \stackrel{\text{def}}{=} (sendCPDk, r_{sendCPDk}).C_7$
$\quad + (sendCAbort, r_{sendCAbort}).C_8$
$C_7 \stackrel{\text{def}}{=} (complete, r_{complete}).C_0$
$C_8 \stackrel{\text{def}}{=} (sendMAbort, r_{sendMAbort}).C_7$

The different states of C component are formulated above. After C performs action *download* in C0, it reaches C1. In state C1, the only action happens is *requestBDigitalCoins* at rate $r_{requestBDC}$ in order to request a digital coin from bank leading C1 to C2, as described in step 2. Then in state C2, there is only one action could happen which is *sendCDigitalCoins* at rate $r_{sendCDC}$ in order to get the digital coin from bank leading C2 to C3, as described in step 3. The states C3 and C4 are same as states C1 and C2 described in pervious protocol model. They reflect step 4 and 5 in the description of this protocol, respectively. Then when C reaches C5, one of two actions happens either *sendMCoinDk* at rate $r_{sendMCDk}$ leading to C6 when C gets valid encryption product or *sendMAbort* at rate $r_{sendMAbort}$ leading to C7 when C gets invalid encryption product M during state C4, as described in step 6. The states C6, C7 and C8 are similar to states C4, C5 and C6 in the basic protocol model, respectively. State C6 reflects step 9 of the optimistic anonymous protocol description.

$TTP \stackrel{\text{def}}{=} (download, r_d).TTP$

TTP has only one state. In state TTP (step 1 in the protocol's description), the only action could happen is *download* at rate r_d leading to the same state TTP.

$B \stackrel{\text{def}}{=} (requestBDigitalCoins, r_{requestBDC}).B$
$\quad + (sendCDigitalCoins, r_{sendCDC}).B$
$\quad + (sendBCoinByM, r_{sendBCByM}).B$
$\quad + (sendMyes, r_{sendMy}).B$
$\quad + (sendMno, r_{sendMn}).B$

The last part of the model is for B component. There is just one state which is B. The B's main actions to support the purchase processes between the components C and M are *sendCDigitalCoins*, *sendMyes* and *sendMno* as described in step 3 and 8 of the optimistic anonymous protocol description. The rates of those actions are controlled by B.

The system equation and complete specification are given by

$\text{System} \stackrel{\text{def}}{=} TTP[K] \bowtie_J (C_0[N] \bowtie_L M_0[N]) \bowtie_R B[S]$

Where the cooperation sets J={download}, L={sendMPO, sendCEP, sendCAbort, sendMCoinDk, sendMAbort, sendCPDk, complete}, and R={requestBDigitalCoins, sendCDigitalCoins, sendMno, sendBCoinByM, sendMyes}, any action in list J, L and M is shared actions between the components specified. N is the number of Cs and M copies on the system, K is the number of TTPs, S is the number of Bs. The four components are initially in state TTP, C0, M0 and B.

Moreover, the service rates of all the main actions carried out by M depend on the number of Cs interacting with M. So, each service rate is divided by the number of Cs interacting with M, as in the basic protocol.

Furthermore, the service rate of all actions of B – *sendCDigitalCoins*, *sendMno*, and *sendMyes* – is dependent on the number of Cs, Ms and Bs. One, two or more Bs can be involved in the protocol to serve C and M. So, each service rate is calculated as follows:

$$r_{\text{sendCDC}} = \left(\frac{r_{\text{sendCDigitalCoi}}}{N}\right) * S$$
$$r_{\text{sendMn}} = \left(\frac{r_{\text{sendMno1}}}{N}\right) * S$$
$$r_{\text{sendMy}} = \left(\frac{r_{\text{sendMyes1}}}{N}\right) * S$$

Where S is the number of Banks and N is the number of Cs and M copies.

3.2.3 *Performance evaluation of the optimistic anonymous protocol*

This investigation seeks to calculate the average response times of M. We assigned 1 as a value for all rates.

The average response times of M1 and M5 in relation to the number of customers is presented in Figure 6, which shows that the average response time of M1 and M5 increased when the number of customers increased. However, the average response time of M5 is larger than M1. We believe this is because M needs to contact B to check the digital coins before response to customer to provide product decryption key. Moreover, the average response times of M1 and M5 are much larger when introduces the anonymous feature for the customer on this protocol compare to the average response times of M1 and M3 of the basic protocol without the feature (Section 3.1.3). M1 and M5 in this protocol similar to M1 and M3 in the basic protocol, respectively. Therefore, adding such a feature to the protocol would add more performance overhead.

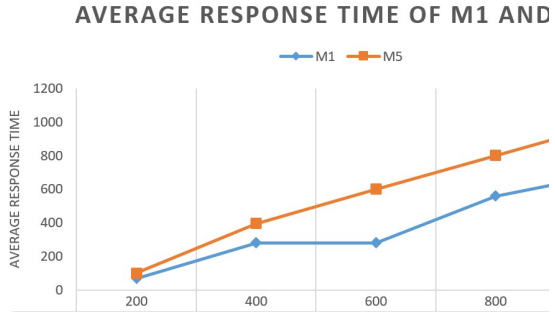

Figure 6: Average response time of M1 and M5 using ODE (*K*=20 and *S*=20).

Moreover, the population level analysis for the average number of customers in C4 and C6 for having a service from M and in C2 for a service from B to get a digital coin is shown in Figures 7, 8, 9 and 10.

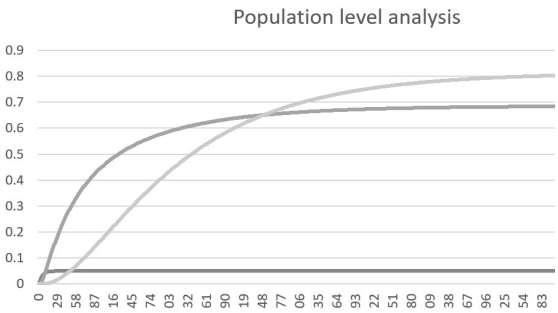

Figure 7: The population level analysis using ODE with *K*=1, *N*=100 and *S*=20.

Figures 7 to 10 illustrate how increasing the number of B involved in the protocol caused the average number of C2 copies to decrease. This suggests that the more Bs are involved in the protocol, the better the performance of the protocol would be. Moreover, more customers are in C6 than C4 for a service form M this is because M needs to check the digital coins with B before sending the product decryption key.

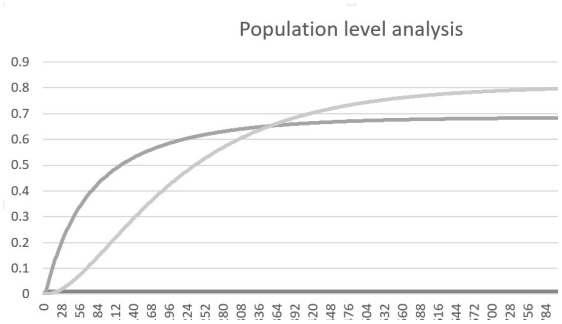

Figure 8: The population level analysis using ODE with *K*=1, *N*=100 and *S*=100.

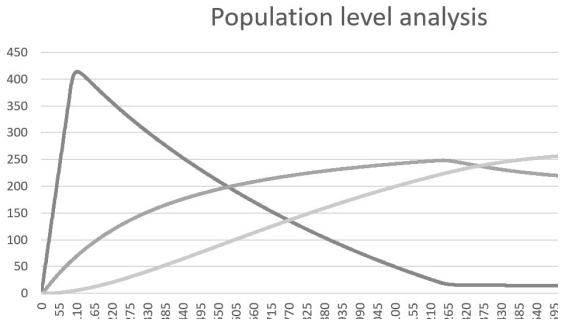

Figure 9: The population level analysis using ODE with *K*=50, *N*=500 and *S*=20.

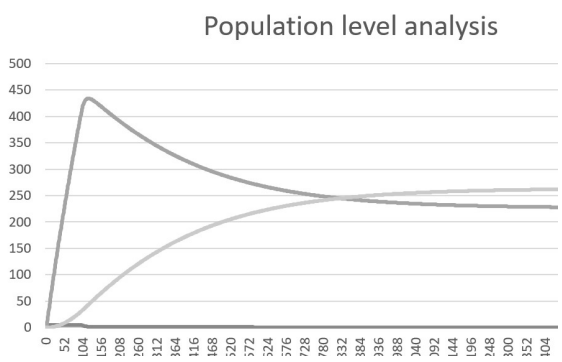

Figure 10: The population level analysis using ODE with *K*=50, *N*=500 and *S*=500.

Figures 11 to 14 illustrate the throughput of sendCAbort, sendCDigitalCoins, sendCEP, and sendCPDk actions. They show throughput values of the main actions to serve the customers in this protocol in relation to different population size of k, n and s. They illustrate how increasing the number of customers would have a significant impact in the actions throughput. The throughput values of action sendCPDk are the lowest values in all graphs compare to other actions. Moreover, action sendCDigitalCoins has the largest throughput values in all graphs. This is because components are moving sequentially from state to state and action sendCDigitalCoins have to be performed first for all customers to be able to perform other actions. Thus increasing the number of B involved in the protocol could improve the protocol performance. Actions sendCAbort and sendCEP have relatively similar values.

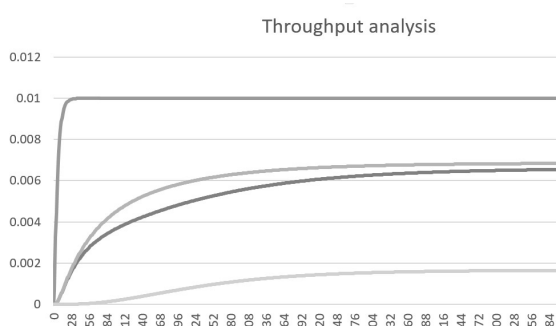

Figure 11: The throughput analysis of actions using ODE with *K*=1, *N*=100 and *S*=20.

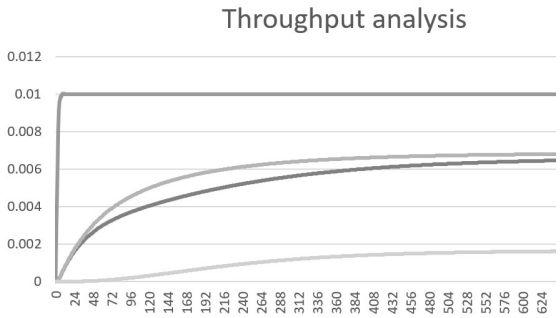

Figure 12: The throughput analysis of actions using ODE with *K*=1, *N*=100 and *S*=100.

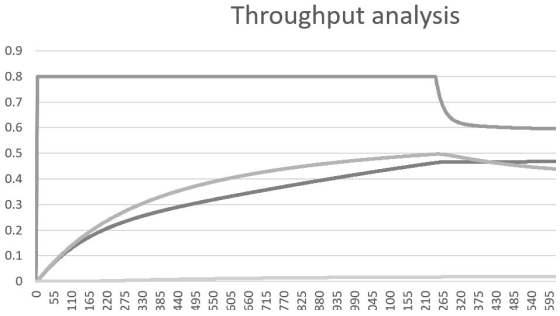

Figure 13: The throughput analysis of actions using ODE with *K*=50, *N*=500 and *S*=20.

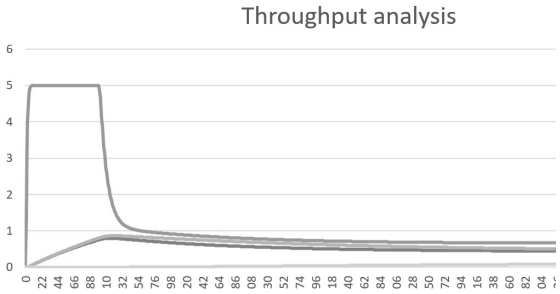

Figure 14: The throughput analysis of actions using ODE with *K*=50, *N*=500 and *S*=500.

4. Conclusions and further work

This work explored the performance cost introduced by a security protocol known as an anonymous and failure resilient fair-exchange e-commerce protocol [10]. Following the description provided by Ray *et al* in [10], the PEPA models were formulated in two different ways: with and without an anonymity feature.

This study used a PEPA eclipse plug-in to support the creation and evaluation of the proposed PEPA models. The results indicated that the basic failure protocol without an anonymity feature introduced lower performance cost than when the protocol preserved the anonymity of customer which introduced extra performance cost.

In our future work, the anonymous optimistic protocol which preserves the customer's anonymity will be modelled when there is dispute between participants so that TTP involvement is active. This will help to achieve a complete and in-depth understanding of protocol behaviour and the associated performance costs. There are a variety of different scenarios that may lead to a dispute, ranging from accidental bad data entry by either party, software bugs, network errors or malicious attempts to defraud. In all cases there will be a load on the TTP to resolve the dispute, but the degree to which this impacts on system performance will vary in intensity. The ultimate goal is to consider 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. In the kind of anonymous optimistic protocol this may mean scaling of TTP resources to handle escalating threats without a negative impact on the rest of the system.

REFERENCES

[1] G. Apostolopoulos, V. Peris, P. Pradhan, and D. Saha, "Securing electronic commerce: reducing the SSL overhead," *IEEE Netw.*, vol. 14, no. 4, pp. 8–16, 2000.

[2] T. Zia, A. Zomaya, and N. Ababneh, "Evaluation of overheads in security mechanisms in wireless sensor networks," in *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*, 2007, pp. 181–185.

[3] K. Wolter and P. Reinecke, "Performance and security tradeoff," in *Formal methods for quantitative aspects of programming languages*, Springer, 2010, pp. 135–167.

[4] C. Lamprecht, A. Van Moorsel, P. Tomlinson, and N. Thomas, "Investigating the efficiency of cryptographic algorithms in online transactions," *Int. J. Simul. Syst. Sci. Technol.*, vol. 7, no. 2, pp. 63–75, 2006.

[5] J.-H. Cho, R. Chen, and P.-G. Feng, "Performance analysis of dynamic group communication systems with intrusion detection integrated with batch rekeying in mobile ad hoc networks," in *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, 2008, pp. 644–649.

[6] Y. Zhao and N. Thomas, "Performance Modelling of Optimistic Fair Exchange," in *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, 2016, pp. 298–313.

[7] Y. Wang, C. Lin, and Q.-L. Li, "Performance analysis of email systems under three types of attacks," *Perform. Eval.*, vol. 67, no. 6, pp. 485–499, 2010.

[8] H. Bossel, *Modeling and simulation*. Springer-Verlag, 2013.

[9] J. Hillston, *A compositional approach to performance modelling*, vol. 12. Cambridge University Press, 2005.

[10] I. Ray, I. Ray, and N. Natarajan, "An anonymous and failure resilient fair-exchange e-commerce protocol," *Decis. Support Syst.*, vol. 39, no. 3, pp. 267–292, 2005.

[11] S. Kremer, O. Markowitch, and J. Zhou, "An intensive survey of fair non-repudiation protocols," *Comput. Commun.*, vol. 25, no. 17, pp. 1606–1621, 2002.

[12] M. Tribastone, A. Duguid, and S. Gilmore, "The PEPA eclipse plugin," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 28–33, 2009.

[13] L. Montecchi, N. Nostro, A. Ceccarelli, G. Vella, A. Caruso, and A. Bondavalli, "Model-based evaluation of scalability and security tradeoffs: a case study on a multi-service platform," *Electron. Notes Theor. Comput. Sci.*, vol. 310, pp. 113–133, 2015.

[14] Y. Zhao and N. Thomas, *Efficient Analysis of PEPA model of Non-repudiation Protocols*. University of Newcastle upon Tyne, Computing Science, 2009.

[15] T. Okamoto and K. Ohta, "Universal electronic cash," in *Annual International Cryptology Conference*, 1991, pp. 324–337.