Design and Evaluate a Fair Exchange Protocol Based on Online Trusted Third Party (TTP)

<u>Ph.D Thesis</u>

Abdullah Shawan Alotaibi

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Software Technology Research Laboratory

De Montfort University

Leicester - United Kingdom

October – 2012

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the software Technology Research Laboratory, at De Montfort University, United Kingdom. No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education

Parts of the work presented in this thesis have been published in the following journals.

Journal publication:

1. A.alotaibi and H. Aldabbas: A Review of Fair Exchange Protocols. International Journal of Computer Networks & Communications (IJCNC). Vol.4, No.4, July 2012
2. A.alotaibi and H.Aldabbas: Design and Evaluate a fair exchange protocol Based on online TTP. International Journal of Security and Its Applications (IJSIA). Vol.4, No.4, July 2012

Abstract

One of the most crucial factors that e-commerce protocols should address is a fair exchange. In this research, an advanced method of cryptography coupled with the pay per use technique is used. A new electronic commerce protocol for the exchange of commodities is introduced. The proposed new protocol guarantees both features while addressing the main drawbacks associated with other related protocols. The new suggested e-commerce protocol is composed of two stages: pre-exchange and exchange stages. When the suggested new protocol is analysed with scrupulous protocol analysis, it attains fair exchange and a secure method of payment. The suggested new e-commerce protocol is more efficient than other related existing protocols. In this research "protocol prototype" and "model checking" is used for the purpose of authentication. The protocol prototype verifies that the suggested new protocol is executable when it's used in a real context. By experimental designs, this research shows the length of asymmetric keys as the biggest element that affects the efficiency of the protocol. When model-checking is applied in this protocol, the outcome indicates that the suggested protocol achieves the required features of fairness. Protocol extensions give those involved in the protocol the capacity to be resilient to failure. By using three methods of authentication, this research confirms that the new proposed protocol is well formulated. The work reported in this thesis first study the existing fair exchange protocols that solve the fairness problem. Then, propose more efficient protocol to solve the fairness problem. The original idea in this thesis is to reduce the communication overheads, risks and solve the bottleneck problems in the protocols that involve an online TTP. The idea is to dividing the process to two phases, pre exchange phase and exchange phase. The proposed protocol has the characteristics: three messages are required between all parties, the protocol guarantee strong fairness for both customer and merchant. The new protocol let the customer to be sure about the merchant’s item before he send his item and let the merchant to be sure about the customer’s item before he send his item, online disputes are resolved by a Financial Service Provider (FSP).

Acknowledgments

First and foremost, my truthful thankfulness goes to the most merciful **ALLAH** for all the things he blessed me with throughout my whole life, without those blessings.

Most importantly, I would like to thank my mentor, supervisor and the one behind this project, **Professor Hussein Zedan**, whom without his support, encouragement and guidance this thesis would not have been possible to achieve. I am so happy that I was able to finish my Ph.D under his supervision. The love and care he offered to his students, including me, has affected this work in so many good ways.

Also, many thanks and gratitude goes to my supervisor **Dr. Feng chen**. For his critical comments, technical suggestions and professional guidance have always improved this thesis since day one.

I want to express my deepest thanks to the one who all the good words would not be enough to describe him, whom without, I would not be the man I am today, to my beloved father **shawan**. What he has done for me from big sacrifices to simple advices is beyond the limits. He was the one to push me forward, to guide me, and to plant every goodness in me. I owe it all to you father. I hope one day I can be the man just the way you are. Also, I would like to thank my mother for her endless support, prayers and guidance since the day I was born. Thank you for being the perfect mother a person can wish for, thank you Mom for everything.

Moreover, my innermost thanks and appreciation goes to my lovely, beautiful and amazing wife. Her love, care, advice, prayers and encouragement have always enhanced my heart, body and soul to the furthest. Thank you for sharing the happy and sad times, the good and bad times with me. Thank you for being you.

I would like to thanks everyone who showed his/her support throughout the years. Last but not least, I would like to thank every member of the **STRL** for providing the academic and home like environment and the support whenever needed, especially **Mrs. Lindsey Trent.**

Tables of content

List of Figures

List of tables

Chapter 1
Introduction

Objectives:

- *Overview*
- *Motivations and research questions*
- *Research Methodology*
- *Measure of Success*
- *Thesis Contribution*
- *Thesis Structure*

1.1. Overview

Due to the rapid growth of electronic commerce in recent years, many businesses are today conducted online. In other words, more businesses than ever before are using the Internet to sell their commodities to people all over the world. The internet provides them with a platform for selling their items to all kinds of people without the restrictions of geographical borders.

Customer choice in buying goods and services has been greatly improved through by this growth in e-commerce, and for a variety of reasons, growing numbers of customers now opt to buy their items through the Internet. Firstly, customers have the convenience of making purchases in the comfort of their homes without having to go shopping malls or having to suffer the hassle of traffic jams. Secondly, customers have the opportunity to quickly compare the prices of various traders. Thirdly, goods and services are delivered to the customer’s home. Lastly, customers are able to buy products at anytime from anywhere in the world.

In traditional commerce, customers do not have to worry that they will be given the product they paid for. This is because the customer goes to a shop, selects a product, pays for it and takes it away. Customers also do not have to worry that their financial data will be revealed to a third party, as they make payment in cash. In addition to the above points, customers can remain anonymous and avoid merchants being able to trace their buying habits through making their payments in cash. In e-commerce, the factors mentioned above vis-à-vis traditional commerce can be major concern for customers. Through online payment, personal data and financial information that are not encrypted might be revealed to fraudsters. There must be trust between the buyer and the seller but in e-commerce, customers may be worried that dishonest dealers will send them the wrong product or not send it at all.

Thus, there needs to be a system in place for ensuring that all data being sent is done so through secure means. There is no doubt that e-commerce has made the exchange of goods and services easier but it also poses risks to both the customer and the merchant, including the issues of security, safeguarding users' privacy, trust and anonymity[112,113]. The main challenge lies in the exchange of the digital commodities between all tips engage in the transaction. This is because the transacting tips do not deal with each other and hence may not confident.

An e-commerce transaction is initiated when a buyer accesses an Internet website through a remote computer. The buyer then places an order for a particular product of his or her choice. If they are assured that the right product will be delivered to them, they formalize the process by placing an order making an electronic payment. It is assumed that the merchant will not disappear into thin air or submit the wrong product after acquiring the payment. This is called fair exchange. Fairness occurs when both parties honour their obligation, i.e. the consumer makes a full payment and the trader delivers the right commodity.

Fair exchange addresses two main problems. Firstly, it ensures that the transacting parties exchange their items fairly. Secondly, it offers an automated online dispute resolution mechanism in case a problem arises between the parties. The protocols of e-commerce should be well formulated in order to avoid dishonest behaviour on the part of any party. Zhou et al [1] have suggested certain terms which electronic commerce should fulfil, and they include the following: (1) The process should guarantee fair exchange; (2) The dispute resolution should be automated and available in case any party misbehaves; (3) The process should guarantee that the products to be received by the buyer are the ordered products, are of the correct quantity, and are free from any damage; (4) The exchange process should involve a trusted third party (TTP); and (5) All the processes of the transaction should be completed, or no process should be completed. Fair exchange protocols are divided into two: the ones that involve the participation of a Trusted Third Party (TTP), and the ones that do not. The TTP guarantees fair exchange of products between the transacting parties. The protocols which use a TTP use the idea of submitting the product in parts. For instance, the customer transmits part of the payment while the merchant transmits part of the product. The exchange process goes on until the complete product is traded.

The use of the TTP in fair exchange protocols are contain three kinds: An inline TTP based protocol the first kind. The role of the TTP is to exchange the respective commodities. In other words, each entity submits its product to TTP, and the TTP then delvers the digital commodity to the buyer and the payment to the merchant. In this case, the TTP guarantees that fairness is achieved in the protocol. The second type of protocol uses an online TTP. The role of the online TTP is only to verify the items to be exchanged, and hence the participation of the trusted third party in this protocol is reduced. The third kind is offline protocols. In these protocols, the transacting entities trade their goods directly, and the TTP only participates to solve proplems; hence, its involvement in the protocol is minimized.

Above, we have discussed the two kinds of fair exchange protocol (those that uses a TTP and those that do not). Their main aim is to guarantee the fair exchange of products for the transacting parties. Protocols that do not use a TTP are much less effective because the process required to complete them is long; in order to obtain fairness for the transacting entities, the protocols should have equal computational power. On the other hand, the major limitation of protocols involve an inline trusted third party is the trusted third party can become the source of a bottleneck but it should always be present. Also, the protocols cannot submit the products to the transacting parties if the TTP crashes.

When we compare protocols that use an online TTP and the ones that involve the use of an inline TTP, we realise that they both have setbacks. The difference is that in online TTP, the participation of the TTP is greatly reduced. Generally speaking, in the offline protocols, the TTP is not use in the protocols unless there is a conflict between the transacting parties.

According to Wang *et al.* [2], human shortcomings and the complex nature of e-systems make it inconceivable to suggest all situations and ensure accurate processing in all conditions, even in well formulated and implemented code. Protocol formulators and participants (stakeholders) must generate ways of eliminating these elusive but ultimately dangerous shortcomings in order to improve control and to safeguard e-commerce users. Such steps are necessary in order to avoid incorrect processing or to thwart fraudulent users who might exploit these weaknesses. To guarantee correct processing in all situations, there is a need for a comprehensive analysis of the methods used in the transaction process. When we consider the methods, model checking is the most appropriate one for analysing e-commerce protocols.

1.2. Motivations and research questions

The most crucial feature of the e-commerce transaction process is fair exchange. Over the last decade, several researches have been conducted in the field of fair exchange protocols. Different protocols have been suggested for different situations vis-à-vis the fair exchange process. The main disadvantage of all the protocols that involve the use of a TTP is that the TTP can become the source of a communication bottleneck, thus resulting in inefficiency in performance. The reason for this poor performance is that all the processes of the transaction pass through the TTP.

The aim of this thesis is to analyse the current fair exchange protocols and address any limitations regarding fairness. We will then suggest a more effective and efficient protocol that will handle and solve the fairness (this means that at the end of the protocol, either each party obtains the expected item from the other or no party obtains the expected item. This means that party that behaves correctly does not suffer a disadvantage. For example, both parties should receive the expected items and none do so) issue[26,112,113]. Our initial aim in this thesis is to resolve the bottleneck issue in the protocols that use an online TTP, and decrease the communication overheads and risks. The second problem is how to confirm that the suggested e-commerce protocol fulfils all the necessary requirements, including fair exchange. Hitherto, most researchers have used informal techniques, such as listing down certain possible situations where attacks from fraudulent users could occur, and then analysing ways of dealing with those problems by using particular techniques.

E-commerce transactions require fair exchange, and this thesis will suggest a new e-commerce protocol that will guarantee fair exchange. In addition, it will also state the properties that have to be fulfilled in the course of the suggested e-commerce protocol. This researcher has analysed the current fair exchange protocols in the literature and we have framed the main research questions as follows:

1. ***How can we design an efficient fair exchange protocol based on an online TTP to solve the increase in communication overheads, risks and bottleneck problems (i.e. improve the overall usability of the protocol)?***

2. ***How can we enhance and increase the fairness property in the fair exchange protocols use an online TTP?***

1.3. Research Methodology

In this research, we have selected a particular scientific research method in order to implement ‘constructive research’. The term ‘constructive’ refers to the knowledge developed as a result of a new protocol, model, method, etc. However, it is difficult to conduct a scientific research in a certain field effectively without sufficient knowledge of the field, and hence, there is a need to acquire meta-knowledge through research.

This scientific research methodology is conducted through four stages. The first stage deals with the research literature review, the second deals with the suggested protocol on which this research paper is based. The third stage explains in detail the formal authentication of the proposed protocol, and the final one assesses the whole work.

- ***Stage 1: Background and critical review.***

  Conduct a critical review of the literature on e-commerce, fair exchange protocols, and classification of those protocols, and provide a standard criterion for fair exchange protocols. Assess the fair exchange protocols in order to authenticate the concept of the suggested protocol.

- ***Stage 2: The proposed protocol.***

  Suggest a fair exchange protocol for digital and physical commodities. Enhance the effectiveness of the suggested protocol.

- ***Stage 3: Formal verification of the proposed protocol.***

  Identify the formal verification systems, particularly model checking, such as Failure Divergence Refinement, Symbolic Model Verifier and SPIN. Make a comparison between the suggested protocol and the other fair exchange protocols in order to confirm their effectiveness.

- ***Stage 4: Implementation and Evaluation.***

  A prototype 'proof of concept' or 'proof of principle' is required to ensure that the new protocol is executable, viable and workable.

1.4. Measure of Success

The success of this research will be assessed as follows:

- The research questions outlined at the beginning must be answered.
- An analysis will be conducted to reveal the difference between our suggested protocol and other protocol types.
- Formulating an effective online fair exchange protocol: various e-commerce protocols have already been designed. This research will critically assess them and formulate a new protocol that will address some of their limitations. Hence, there will be a comparison between the new protocol and the relevant fair exchange protocols.

- Specification of the effectiveness of the suggested protocol: the new protocol will be specified; we will indicate the number of messages required, and the subject matter of these messages will also be specified.
- An automatic built-in dispute resolution mechanism: there is the possibility that conflicts between the transacting parties will emerge at one time or another. Such disputes will resolve in a judgment. The new protocol is intended to decrease the number of disputed cases but we will also provide an automatic dispute resolution mechanism.
- The protocol should ensure strong fairness for the transacting parties.
- A proof of concept implementation: a prototype proof of concept is required to ensure that the new protocol is workable, viable and executable.

1.5. Thesis Contribution

This thesis has made a new contribution by formulating a fair and secure protocol for the exchange of digital commodities and payment. The thesis has also introduced a trusted third party, FSP (Financial Service Provider) that participates in the exchange of the digital product between the transacting parties. This is a well formulated new protocol. The validity of the proposed protocol is authenticated to ensure that the protocol is correct, strong and fulfils all specifications. The introduction of a third party, FSP, has raised security issues which have been addressed sufficiently in order to ensure that the protocol is strong and free of error.

1. This research has achieved a new e-commerce protocol that insurer’s fairness and customer anonymity for both parties involved in the transaction. Apart from the role of the customer and the merchant, this new protocol clearly stipulates the role of the trusted third party, FSP, in the exchange process.
2. Since customers usually use the symmetric encryption and decryption during the exchange process, they can reduce their computational overheads.
3. In order to authenticate the suggested e-commerce protocol, this research has used three different authentication methods: These methods are: protocol analysis, protocol prototype and model checking. In general, the new protocol discussed in this research is a well formulated and secure protocol that ensures fairness for the two parties involved in the transaction process.
4. Only three messages are required to be exchanged between all parties.
5. The protocol guarantees strong fairness for both customer and merchant.
6. The new protocol let the customer to be sure about the merchant’s item before he send his item and let the merchant to be sure about the customer’s item before he send his item,
7. Disputes are resolved automatically online by a Financial Service Provider (FSP).
8. Reduce the communication overheads and solve the bottleneck problems.

1.6. Thesis Structure

The following is a brief outline of the remaining chapters of this thesis, together with a summary of their contents.

- ***Chapter 2: E- Commerce***

In this chapter, an introduction to e-commerce is presented. E-commerce advantages, disadvantages and limitations are discussed, and some important issues are raised (such as the security of e-commerce and e-payment).

- ***Chapter 3: Fair exchange protocols***

What is the fairness and fair exchange protocols are discussed in this chapter. Dispute resolution is also covered and some of its techniques are presented. In this chapter, we discuss some of the concepts behind the designs and characteristics of protocols. Also, some of the cryptographic concepts are discussed.

- ***Chapter 4: Our proposed protocol.***

This chapter discusses the concepts behind our protocol's design, details the rules of design and describes its properties; some general assumptions for the suggested protocol are also clarified. The new protocol is then presented, studied and contrasted with other fair exchange protocols.

- ***Chapter 5: Formal verification.***

This discusses the formal authentication of systems, specifically model checking, in order to authenticate the fairness property; model checking will be used to model the protocol.

- ***Chapter 6: Protocol Implementation***

A prototype proof of concept execution is formed using the Java programming language. The design and execution of the prototype that executes the protocol is also discussed.

- ***Chapter 7: Conclusion and Future Work***

A summary of the work in this thesis is presented in this chapter, and the results are summarized. Then we write the conclusions and suggest ideas for future work.

Chapter 2
Electronic Commerce

Objectives:

- *Advantages of E-commerce*
- *Process of E-commerce Operations*
- *E-Payment Systems*
- *Characteristics of Electronic Payment Systems*
- *E-Payment System Models*
- *E-Commerce Security*
- *E-commerce Trust*
- *E-commerce Limitations*

In this chapter, an introduction to e-commerce is presented. E-commerce advantages, disadvantages and limitations are discussed, and some important issues are raised (such as the security of e-commerce and e-payment).

2.1 Electronic Commerce

Electronic commerce, also called e-commerce or e-comm, can be known as the purchasing and selling of goods and services electronically over computer networks such as the Internet. E-commerce has developed from technologies such as e-funds transfer, supply chain management, e-marketing, e-transaction processing, e-data interchange (EDI), inventory management systems, and automated data collection systems. The World Wide Web is the main channel of transaction in e-commerce, although it may include other forms such as cell phones, e-mail and telephones. Basically, e-commerce is taken to be the sales part of e-business. In addition to sales, it encompasses exchange of information in order to enhance the funding and payment forms of business transactions [3].

The main goal of a customer is to acquire a product that is in the hands of a merchant, while the main goal of the merchant is to sell the product to the customer and obtain payment from that customer. Although the services offered by the merchant to the customer, such as ease of transaction and after-sales service, are crucial, the main concern for the customer is ensuring the delivery of the product ordered. Hence, the transacting parties exchange their items, which are a payment and a product, in a fair manner. This means that, at the end of the exchange process, the transacting parties should have each other's items or none of them do. This method of trading items fairly is known as a fair exchange protocol [4].

The following are the different kinds of e-commerce [3]:

- ***Business-to-Consumer (B2C):*** an online transaction that is carried out between a business and individual consumers.
- ***Business-to-Business (B2B):*** this refers to online businesses transacting with each other, such as suppliers selling to distributors.
- ***Customerr-to-Business (C2B):*** this point to personals selling their products or services to companies.
- ***Consumer-to-Consumer (C2C):*** this refers to a person-to-person transaction where individuals can buy and sell online directly to each other.

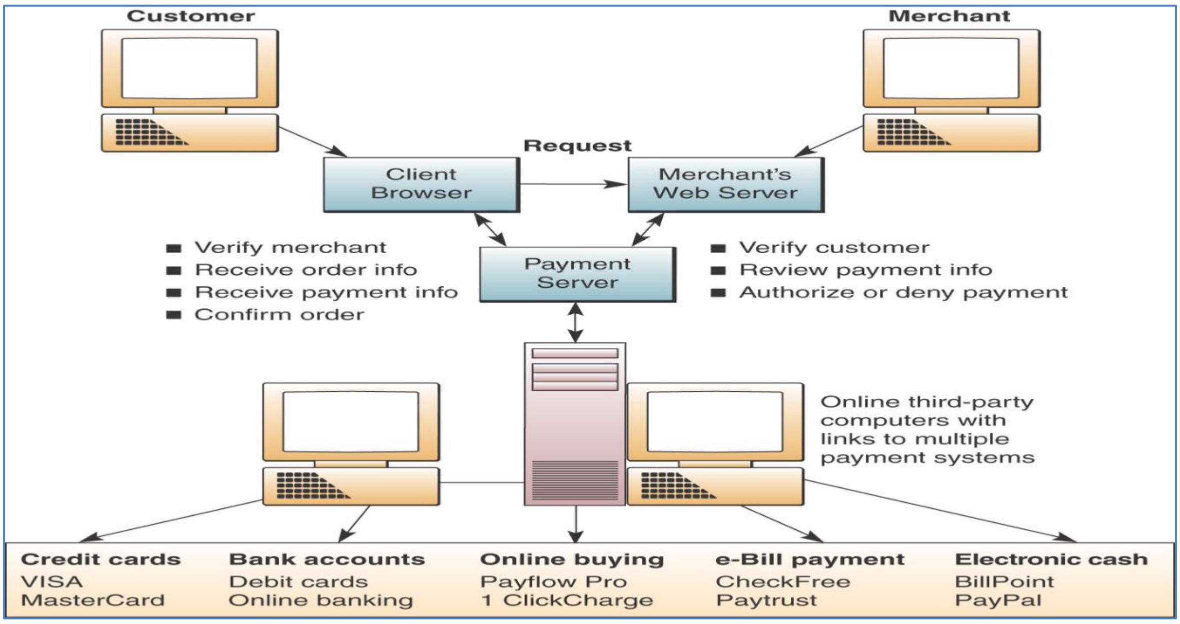


Figure 1: Business-to-consumer [6]

Figure 1 show an example of B2C e-commerce, where a merchant (business) and a customer (consumer) are transacting with one another. However, the processes shown in Figure 1 can be applied to all four types of e-commerce, in that the customer first searches the web for a product, places an order and makes the payment, while on the other hand the merchant sends the product and provides after-sales services. Thus, all four types of e-commerce follow the same process during a transaction.

2.1.1. E-commerce Advantages

- ***Shopping from home:*** customers are able to purchase products from the comfort of their home, without the hassle of traffic jams and standing in queues, by just going online.
- ***Shopping at any time:*** e-commerce has no time limit, as customers can purchase products at their leisure any time of the day or night. This is also an important factor for merchants in e-commerce, as they can sell their products to customers at any time because their websites have no closing and opening time.
- ***Cheaper prices:*** customers can buy cheaper and better-quality products through the Internet. This is because e-commerce reduces the costs of running a business, unlike traditional commerce where more employees need to be engaged, leading to higher running costs.
- ***Home delivery***: e-commerce often provides free delivery to the customer's home, particularly to those who spend over a certain amount.
- ***Online sales support***: customers can access a big amount of information about the product on the e-commerce website, and hence make a well-informed choice.
- ***Global reach:*** e-commerce enables merchants to reach the global market, as business activities are not restricted by geographical boundaries; merchants can sell their products in both the national and international market.

- ***The number of employees is reduced:*** most activities in e-commerce are automated and hence fewer employees are required to carry out the operations.
- ***Instant delivery:*** if the customer purchases an e-product (like e-book or a movie) online, then it is electronically sent to the customer immediately through a link in the merchant's website (or by e-mail).
- ***Shopping for all people:*** all kinds of people can purchase online products, even those who live in remote areas or have special needs, provided they have an Internet connection [4, 5].

2.1.2. E-commerce Disadvantages

Despite the many advantages mentioned above, e-commerce is characterized by certain disadvantages, which can be summarized as follows.

- ***Privacy and security:*** these two issues are very important to both parties involved in the transaction. Customers are concerned about the security of their personal and financial information when they are purchasing items online. On the other hand, merchants are worried about the security of their website because if it is not well protected from dishonest users, their reputation may be tarnished.
- ***Delivery:*** sometimes, customers are kept waiting due to delays in the delivery of the expected items. Other times, the wrong product might be sent to the customer, or in some cases no product at all is delivered, particularly if the merchant is fraudulent.

- ***Inspecting products:*** in traditional commerce, customers are able to inspect and feel the product but in e-commerce, consumers are not able to see the actual product. Instead they are only shown a picture and a description of the product.
- ***Social interaction:*** in traditional commerce, there is personal deal between the consumer and the merchant, while in e-commerce, the relationship is completely impersonal. This is because in e-commerce most of the transactions are conducted by computers. This is likely to create some social problems in the long run.
- ***Returning products:*** returning an item and asking for a refund can be very troublesome and time consuming in e-commerce, particularly if the transacting parties live in different geographical areas [4,5].

2.1.3. E-commerce Limitations

According to some writers, e-commerce limitations can be divided into technological and non-technological limitations [4, 5 and 7]. These limitations are reviewed in the following sections.

2.1.3.1. The technological limitations

Technological limitations refer to those that are inherent in the technology that is used in e-commerce. The following are some of these limitations:

- There is a lack of universally accepted standards for quality, security and reliability.

- Internet connections still remain expensive or out of reach for many users.
- There is a lack of universally agreed upon standardized quality-measurement methodologies.

2.1.3.2. The non-technological limitations

Non-technological limitations refer to the risks and problems that users encounter when using e-commerce. The following are some of these non-technological limitations[4]:

- Security and privacy issues are the major concerns for online shoppers. This is because the Internet is vulnerable to fraud and other abuses.
- Most people do not have trust in e-commerce because they fear disclosing their personal and financial information for security concerns. Others do not trust the technology itself and this reduces the number of potential online shoppers accordingly.
- The rise in Internet fraud has led some customers to stop using e-commerce.

2.1.4. Process of E-commerce Operations

Before we classify e-commerce, we must understand the entire transaction process. There are three main stages involved in the e-commerce transaction process:

- The searching stage: this is the stage where the customer or the merchant browses the Internet.
- The ordering and payment stage: this is the stage where the customer places an order for a particular item, and then makes the necessary payment after an agreement has been reached.
- The delivery stage: in this stage, the ordered products are delivered to the customer.

The modern technology allows the first and second stages of the above transaction process to be done online. The customer browses the Internet, chooses and/or makes enquiries about a product or service, and pays for it electronically.

When we consider the third stage, only digital products and services can be sent electronically; physical commodities such as industrial and agricultural products cannot be sent online. We can only send products that can be changed into digital form through the Internet. The diagram below gives a clear illustration of the three main stages of the e-commerce operations, where the buyer first browses the Internet, selects an item, places an order and makes the payment (all through the Internet). Then, the seller delivers the ordered product or services to the customer either physically or online, depending on the kind of product, as illustrated in Figure 2.

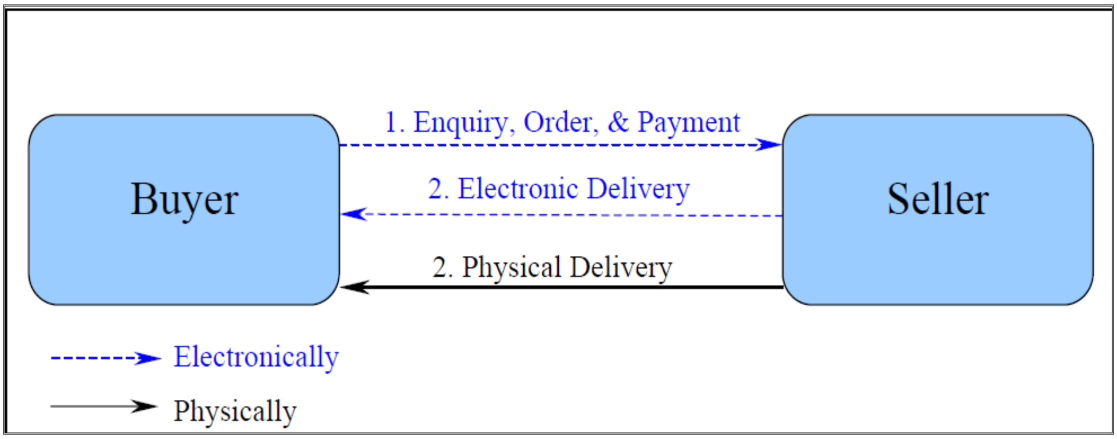


Figure 2: The process of e-commerce operations [7]

Digital products that are sold and downloaded from the Internet have no international barriers. Such products can be sent electronically from one nation to another without any restrictions. This is because it is very hard to measure such a transaction and to place restrictions accordingly. However, when delivering physical products, there are international barriers that cannot be waivered [7]. This is clearly illustrated in Figure 3.

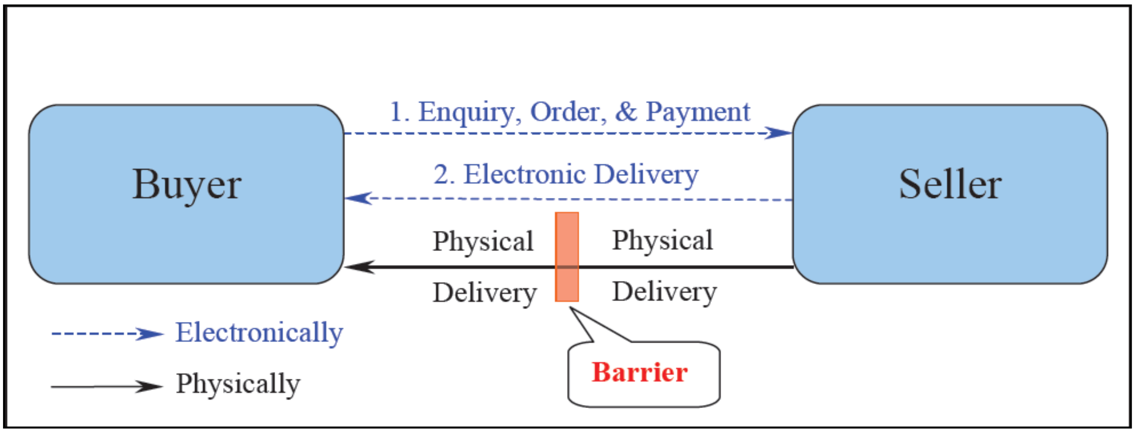


Figure 3: The process of international e-commerce operations [7].

2.2 Electronic Payment Systems

In e-commerce, an electronic payment system is used instead of coin- and paper-based cash. The system allows buyers to make electronic payments for the products that they buy online. Other systems used by banks, such as bank draft and cheque, also have the same purpose. Hence, e-commerce payment eliminates the need to carry money. However, safe and secure transaction is a critical characteristic of any e-commerce payment system in this Internet era, where online deception and fraud is a very common occurrence.

The operations of electronic payment systems happen very quickly. Credit cards are most often used for purchasing items over the Internet, however, credit card users are highly sceptical about the safety of their funds and the security of the Internet because of widespread Internet fraud and scams [8], which can result in financial losses for the buyer and the seller, and for the bank that is involved in the transaction. Examples of older systems that use electronic methods are: the electronic clearing system (ECS), check transaction system, online credit card transaction system, etc.

However, the current electronic payment systems are not perfect due to the higher fixed-transaction costs, Internet scams and the simultaneous participation of several parties in the payment process. These systems have no standard compatibility, and lack user confidence and proper application plans. Modern electronic payment systems should live up to the high expectations of users and merchants.

The current e-payment systems can be categorized into: electronic cash and credit/debit systems [9] and the account-based and token-based systems [10]. Tokens and e-cash are similar to normal cash, in that they represent value but the credit/debit and account-based systems do not embody value, rather a message to send value.

2.2.1 Features of Electronic Payment Systems

The success or failure of an electronic payment system is determined by several factors or features. These include trust, security, user friendliness, interpretability, traceability, etc. Abrazhevich in [10] Gennadey Medvinsky and B. Clifford Neuman in [11] have divided these features into two features relating to the user, and features relating to technology.

However, we argue that, although informative, this distinction is a little simplistic as there is a degree of overlapping between technology and user accordingly we describe these features thus:

- Applicability: the system should accept the user if he uses right procedure to purchase items or services.
- Simple: the system should be user friendly. People in remote areas of the world should be able to use it.
- Security: this deals with the implementation of the value (money). Creation, alteration and over-spending of that value must be safeguarded.
- The integrity of the value and the mandate for the value should be deemed satisfactory on the part of the concerned users.
- Reliability: the system should run smoothly and the probability of failure should be low.
- Trust: the user should have confidence in the system, knowing that their finance and personal data will not be compromised.
- Scalability: the system should function well even when the workload changes in size and volume.

- Convertibility: the currency or value should be convertible from one form to another.
- Interoperability: this refers to the ability of the system to work with other systems.
- Efficiency: micro-payments should be cost effective in the sense that they should not be inordinately expensive.
- Anonymity: confidentiality should be guaranteed in order to protect the privacy of the user.
- Traceability: the system should be able to rack the money in the system, such as who sent what and when it was sent, without compromising the anonymity of the user.
- Authorization type: both offline and online transactions should be secure, and the procedures should be the same.

2.2.2 Electronic Payment System Models

When purchasing goods or services online, the buyer sends the payment to the merchant electronically. In the traditional system, both the consemer and the merchant physically exchange the payment and the product, and in this form of commerce, there is no participation of any third entity in the transaction.

However, for an electronic payment system, several models have been proposed by different organizations and researchers, and some of these systems are briefly discussed below.Ahmad-Reza Sadeghi and Markus Schneider classified e-payment systems into four kinds [11]: e-cash, cheque or credit card, and remittance and debit order-based systems. In e-cash-based transactions, the customer withdraws his/her e-cash or e-token from the bank where he/she holds an account and the bank debits that buyer's account by the value equal to the amount of that token. The customer after that buys items using the e-cash or token. After receiving the e-cash, the merchant deposits it in his/her own bank account. The merchant's bank then submits an order to the customer's bank in order to transfer the money and deposit it in the merchant's account.

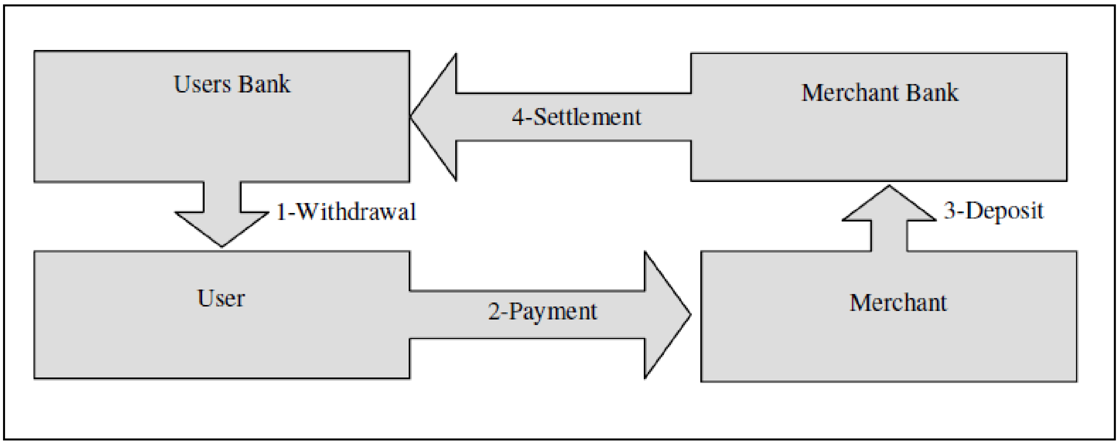


Figure 4: Electronic cash-based payment model [11].

There is no withdrawal for the user in the cheque and credit card payment system (level 1 in the figure above). The merchant merely deposits the cheque or credit card slip in his/her bank. The buyer's bank then send the money into the seller's account.

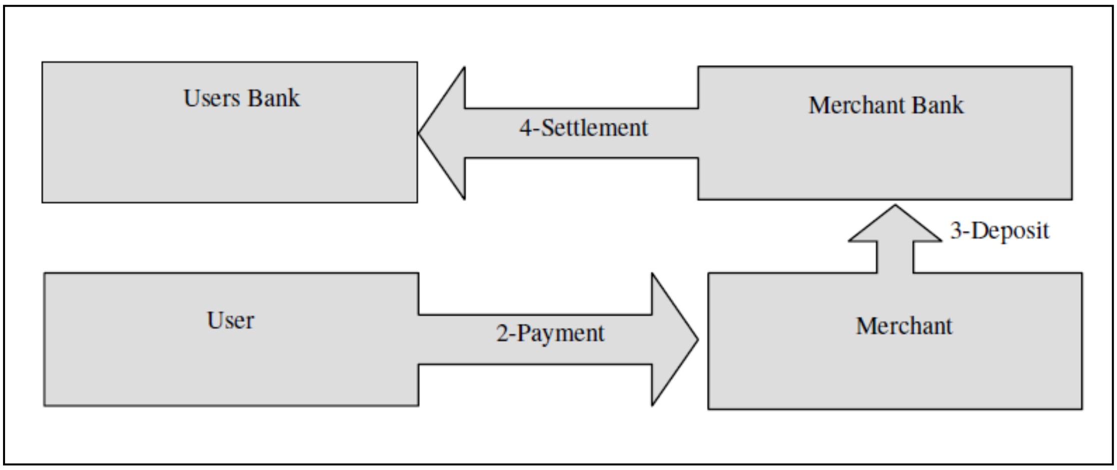


Figure 5: Cheque/credit card-based payment model[11]

In the other two types of electronic payment systems, both the user and merchant instruct their respective banks to transfer the money. Terminologies such as 'issuer' for the user's bank and 'acquirer' for the merchant's bank were introduced by N. Asokan.

2.3. E-Commerce Security

The issue of security is a major factor in e-commerce. As a result, most customers opt to purchase goods and services from well-secured websites and trusted merchants. Therefore, in order to protect the private and financial information of the customer and to motivate them to shop online, merchants must invest a great deal in protecting their systems from fraudulent access. US$ 6.2 billion was spent on security around the world in 1999 but this reached $25 billion in 2002 [13]. This shows that merchants recognize the increasing security threats of e-commerce and are now spending huge sums to overcome them. E-commerce systems are prone to fraud, misuse and failure [14], and the effects of these security threats on e-commerce impact on both transacting parties. Some of these effects are outlined below.

Firstly, both transacting parties can lose huge sums of money. In 2000, established e-commerce pages (such as eBay, Buy.com and tesco.com) were attacked, and these led to losses estimated at approximately $1.7 billion [13]. Secondly, classified data such as debit or credit card information are prone to theft. A case in point is that the TK Maxx stores were attacked in 2007, and private information was stolen from 45.7 million payment cards (debit and credit) [15]. Thirdly, online shoppers will lose trust in e-commerce if there is lack of security in the system; some may cease shopping online altogether. Finally, unauthorized users might take advantage of the resources in the system.

The above effects demonstrate the urgent need for a safe and secure e-commerce system. If the e-commerce system is safe and secure, more customers will shop online, hence leading to significant increases in sales and profits for merchants. E-commerce systems are threatened by two types of attack [13]; the first is non-technical. This kind of attack is carried out by people who work within the system. They might reveal confidential information to unauthorized persons or carry out activities that compromise the security of the system. The second type is technical; this refers to an attack on the infrastructure of the system, and it is performed by software. Examples of technical attacks include computer viruses, worms and Trojan Horses [4].

In order to achieve secure e-commerce systems, three issues need to be addressed [8], and these are:

1) *Customer security*

If customers use unsecured software (such as Internet browsers) when buying goods and services over the Internet, they may be responsible for undermining the safety of e-commerce.

2) *Data transport security*

When customers use an e-commerce system, they make payments or send their private information online. There is a need for a secure system in order to protect these private data and to prevent dishonest users from accessing them. The following are some of the methods that can be used to safeguard the privacy of these data.

- Public key infrastructure (PKI): this consists of SK and PK encryption, digital signatures and certificates.
- Secure socket layer (SSL) protocol [SSL]
- Secure E-Transaction (SET) protocol [SET].

3) *Merchant security*

It is crucial to secure the merchant's system, as it contains highly classified information (for both customers and merchants). Consequently, failure to secure the merchant's system may result in customers losing trust and faith in the merchant.

In summary, an e-commerce system is used to conduct the purchasing and selling of items (whether digital or not) between a customer and a merchant via the Internet. Therefore, the e-commerce system should be well secured in order to protect sensitive information (both the customer's and the merchant's) from unauthorized access.

The customer's side should be well secured in order to protect the personal data and financial information of the customer. On the other hand, the merchants' side should be secured in order to protect the customer's data and the product to be sent. Also, the communication channel between the transacting parties should be well secured in order to protect the items being sent between the parties.

The customer's side can be protected by using secured software. The merchant's side can be protected by using a secured web server and a secured operating system for the network server. Lastly, the communication channel can be protected by using strong security infrastructures and protocols such as SSL, SET and PKI. Another security threat is the issue of protecting both of the transacting parties from each other. For example there is a need to protect fraudulent merchants from honest customers and vice versa.

2.4. E-commerce Trust

Building trust and credibility is an important factor in business. The issue of trust is of particular importance in e-commerce because online shoppers have limited knowledge about the merchant and the merchant's ability to deliver the ordered goods. Trust and security are considered to be the most crucial factors in the success of e-commerce [16]; most customers who are fearful of shopping online state that they lack trust in the merchants [17]. 45% of agents interviewed in a survey of 60 agents at a US company stated that they avoid shopping regularly online Because of the loss of confidence in the system [18].

Increased customer confidence in online shopping can increase sales. If trust is established between the transacting parties, then each of them is likely to fulfil their obligations. For example, if the customer has confidence in the merchant, he/she will make the payment in good time with the expectation that they will receive the ordered product in a speedy manner.

Two kinds of trust in e-commerce exist; the trust in the technology used and the trust in the trading partner. Technological trust can be achieved through the reliability of the system; unreliable technology can affect the trust of customers in the e-commerce system [20]. It is very difficult to establish trust with a partner but it is easy to lose, and this can lead to serious cost consequences [17].

Trust is normally built over time through honesty and integrity. In the case of e-commerce, the customer must interact with the merchant over a period of time in order to develop trust and vice versa. The same applies for the merchant because there are customers who are trusted by the merchant but there are others who are dishonest and are untrustworthy. Previous interactions between the parties play a significant role in building trust. Good relationship with other people can also build trust, i.e., if a customer has a good reputation with a trusted merchant, then this customer is likely to recommend the merchant to other customers. Generally, trust can be earned by having a good reputation.

According to Srinivasan [21], there are five factors that promote trust between the transacting parties in e-commerce:

1. Easy accessibility to the display and description of goods and services available in a merchant’s website.
2. Simple procedures of ordering items.
3. Customers should receive order confirmation after placing an order.
4. Customers should be able to track the items they have ordered.
5. An offer of after-sales services to customers.

The above factors are very significant part in construction the consumer confidence and trust in purchasing products from a merchant, particularly when the customer is interacting with the merchant for the first time. After interacting with the merchant and gaining experience, the customer will be confident when buying items on subsequent occasions and may even recommend others to buy products from the merchant. In general, customer confidence and trust can be established by past experience combined with other factors such as good reputation [4].

There are various ways of improving trust and confidence between the customer and the merchant in e-commerce. For example, e-Bay [22] allows customers to make comments after purchasing a product from a merchant through a manual process. These remarks reflect the buyer’s (customer’s) opinion about the product sold by the merchant, and can be viewed by anyone who visits an e-Bay website. These comments will help merchants to improve their reputation and will encourage more customers to purchase goods from them. The notion of improving the level of trust used by e-Bay can be called a ‘trust profile’ for a merchant, and the same can be done for customers. For example, a merchant can keep and view the trust profile of a customer when they are making purchases in order to identify their trustworthiness by looking at their past purchasing habits. The trust profile is done manually and there is scope for automation, probably by using fair exchange protocol or some other system. Thus, trust between the transacting parties is a critical issue in e-commerce. Confidence and trust in the partner involved in the transaction, whether customer or merchant, will either motivate or demotivate the user to participate in the e-commerce system. As a result, it is vital for the success of e-commerce to formulate e-commerce protocols that can improve trust between the transacting parties. Regardless of fraudulent users (customers and merchants), e-commerce protocols should guarantee fair exchange of payments and digital products between the parties involved in a transaction.

2.5. Summary

The two parties involved in e-commerce transactions normally exchange items. The items that are exchanged are a payment on the part of the customer and a product on the part of the merchant. The objective of the customer is to acquire the ordered products, while the objective of the merchant is to acquire the correct payment. This process of exchange is referred to as the fair exchange of goods.

A general description of electronic commerce was presented in the introduction to this chapter, and the advantages, disadvantages and limitations of e-commerce were outlined. The major threat facing e-commerce, which is security, was also discussed. It was shown that in order to achieve a secure e-commerce system, three aspects must be secured. They are the software that is used by the customer, the channel of communication that is used to transmit and receive the data between the transacting parties, and lastly, the server on the merchant’s side. It was also discussed in this chapter that trust of dealing in exchange items is critical to the expansion of e-commerce. E-commerce users want to trust both the technology of the e-commerce system and the partner with whom the user will interact. Trust in a partner is normally gradual and it is difficult to build but easy to lose. Trust between the transacting partners can be built by using protocols that guarantee fair exchange of payments and products. Lastly, this chapter reviewed e-payment systems, in particular those that involve the use of a credit/debit card, by giving illustrations on how information is exchanged between the consumer, the seller and the bank.

Chapter 3

Fair Exchange Protocols

Objectives:

- *Definition of Fairness*
- *Fairness in Electronic Commerce*
- *Overview of Fair Exchange Protocols*
- *Dispute Resolution*

Definitions for fairness and for fair exchange protocols are presented in this chapter. Dispute resolution is covered and some of its techniques are presented.

3.1. Definition of Fairness

Fairness is a broad concept, employing various terminologies that are adaptable to different fields of application. The term fairness, which has lately been introduced into e-commerce, refers to an impartial or unbiased exchange of items between the transacting parties in such a way that no one gains advantage.There are many types of fair exchange protocols exist,which formulated for to grantee fairness in e-commerce. Each of these protocols has a different idea of fairness, and hence it is quite difficult to compare or formally validate them, as there is a lack of any unified or standard formal definition for fairness [23].

Thus, the term fairness has a number of definitions; there are more than nine entries for the definition of the adjective ‘fair’ in The Free Online Dictionary (by Farlex)[24]. Accordingly, the concept of fairness is used in academic circles apparently in varying although distinct ways. Basically two somewhat different linguistic definitions linked to the usage of the term fair in computer science:

1. The first definition is characterized by conformity to generally recognized standards of propriety or morality.
2. The second definition relates to the process of applying ‘the rules’ equally to all concerned parties and items (this definition is different from the first). The main goal of e-commerce is to provide a platform that enables technologies [25] to provide services (buying and selling goods) online in addition to performing other functions such as advertisement and maintenance.

It can be inferred from these that the second definition has been adopted in e-commerce. The basic issue to be addressed in e-commerce relates to the fair exchange of goods between the transacting parties. There have been formulations of different kinds of fairness; however the problem is that their meanings have remained informal. Lack of formally universally accepted definitions of these terms will make it difficult to validate and compare fair exchange protocols. Hence, there is a need to formalize the meanings of these terminologies.

3.1.1. Fairness in Electronic Commerce

According to Asokan [26], a fair system refers to a system "that does not discriminate against a correctly behaving player. As long as a player behaves correctly, a fair system must ensure that other players will not gain any advantage over the correctly behaving players." In a fair exchange scenario, the transacting parties, for example X and Y, follow a fair exchange process. This process must not allow a situation whereby X can receive Y's items while Y cannot receive X's items. A process that involves a fair exchange protocol between X and Y must fulfil three conditions:

1. ***Effectiveness:*** if the protocol is executed correctly and the parties X and Y honour their commitment, then both parties will have each other's items.
2. ***Timeliness:*** the protocol will be finally executed [112,113].
3. ***Fairness:*** there are two types of fairness:
    - ***Strong fairness:*** this means that at the end of the protocol, either each party obtains the expected item from the other or no party obtains the expected item. This means that party that behaves correctly does not

suffer a disadvantage. For example, both parties should receive the expected items or none do so.

- ***weak fairness:*** this means that at the end of the exchange, either strong fairness is achieved, or the correctly behaving party that does not receive the expected item can prove to a third party that Y has received (or still can receive) X’s item without any more involvement from X (regardless of whether Y behaves correctly or not), and vice versa. Although strong fairness is desirable, sometimes it is very expensive or impossible to guarantee, and that is why the two forms of fairness exist.

Weak fairness is important because it provides a platform for a dispute resolution; the disadvantaged party can seek a dispute resolution outside the system. The party that suffered a disadvantage can achieve strong fairness by using an external dispute resolution system, such as a court of law, provided it can prove that it was treated unfairly. A number of fair exchange protocols ensure strong fairness by using a trusted third party (TTP). Most of these protocols, apart from Burk and Pfitzmann [27] refer to the fairness definition of Asokan [26,112,113].

3.2. Overview of fair exchange protocols

The main reason why we require protocols in e-commerce is to manage the transaction between the buyer and the seller. In order to motivate potential customers into engaging in e-commerce, protocols should be well formulated and secured. These protocols will protect both parties from fraudulent users and subsequently promote the growth of e-commerce. There are protocols which are designed to guarantee fairness between the customer and the merchant so that neither party gains advantage over the other [112,113]. These protocols are known as fair exchange protocols.

A trusted third party (TTP) is a nonpartisan party (entity) or an impartial intermediary used in fair exchange protocols, whose role is to ensure that each party receives the item it expects, or none do. It is assumed that the TTP is neutral, available and trusted by all groups. Sometimes, more than one TTP might be involved in a transaction. Accordingly, the TTP carries out all or some of the roles shown below [28,112,113]:

- Ensures fair exchange of items.
- Acts as an agent of delivery, for example, gives items to the concerned parties.
- Acts as a reliable and trusted agent for the transacting parties.
- Solves problems between the parties in case of disputes.
- Validates items and awards certificates.

Fair exchange protocols ensure that the two parties involved in the transaction exchange their items fairly. Often transactions occur between parties who are not familiar with one another and (or) may not trust one another. To facilitate fair exchange and protect both parties, fair exchange protocols have been designed, and their objective is to ensure that, at the end of the exchange, both parties receive each other's items, or none do.

The current fair exchange protocols may be classified into two main types, depending on the use of the TTP. The protocols that do not involve the use of a TTP are the first type, while the protocols that do involve the use of a TTP form the second type [29, 30, and 31]. The protocols that involve the use of the TTP can be divided into three sub-types, which are as follows [32]:

- Inline protocols depend on a trusted third party.
- Offline protocols depend on a trusted third party.
- Online protocols depend on a trusted third party [112,113].

• *3.2.1. Inline protocols depend on a trusted third party*

Such as in [27, 33, 34], the TTP is for sending the traded commodities to the respective parties. This means that the TTP receives the items from each party, authenticates them and delivers them to the respective parties. For example, if there is a customer and a merchant in a transaction, then the two parties will exchange items such as a digital product (held by the merchant) and a payment (held by the customer). The protocol is then carried out in the following manner. Both the customer and the merchant send their items to the TTP.

The customer sends the payment while the merchant delivers the digital product. Then, the TTP authenticates the received items and, after approving them, delivers the payment to the merchant and the digital product to the customer [112,113].

Figure 6 illustrates a model of the fair exchange protocols that involve an inline TTP. We see in this protocol that the TTP is involved actively in the exchange of items between the transacting parties. Involving the TTP in this type of protocol guarantees that the parties involved in the transaction exchange their items fairly. Direct contact between the transacting parties is not normally necessary in the inline TTP-based protocols [112,113].

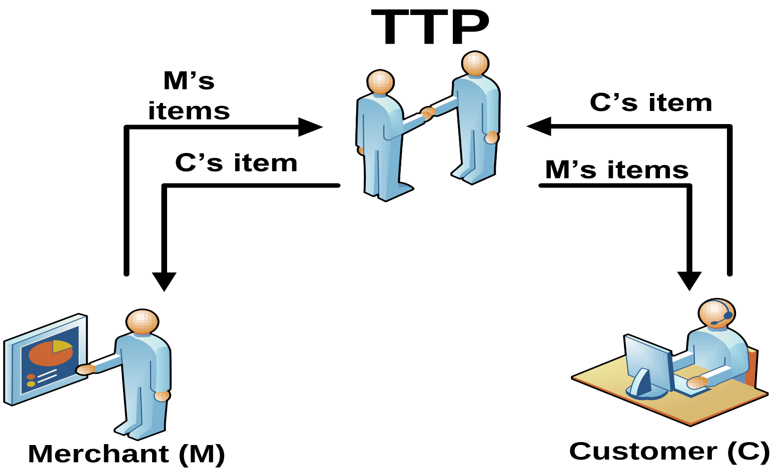


Figure 6: Inline TTP-based fair exchange model

The protocols that use inline TTP guarantee fairness for all parties involved in the transaction because the TTP delivers the respective items to the parties, however, they have some drawbacks. Firstly, it is expensive to run inline TTP protocols because they require the availability of the TTP during the execution of the protocol, which lead to extra costs [39]. Secondly, in this type of protocol, the TTP may become the source of a communication bottleneck, hence leading to performance problems [35, 36, 37 and 38]. This is because the items to be exchanged must pass through the TTP. Thirdly, in case there is a crash in the TTP, the protocol will not be carried out and the parties will not be able to receive the items that they expect. The actual real life example of this kind of protocols is PayPal.

Lastly, in case of any attack, the TTP will be the main target [36,112,113]. Examples of this kind of protocol are in Appendix A.

• *3.2.2. Offline protocols depend on a trusted third party*

Such as in [33, 35, 37, 40, 41, 42], the transacting parties exchange their commodities directly without the use of the TTP unless a problem occurs. This type of protocol is also known as in the literature ‘optimistic fair exchange.

The example below explains how the optimistic fair exchange protocols work if the commodities to be traded between the transacting parties are a payment and a digital product. The two parties directly trade their items, and in case of any problem, the TTP will be involved to mediate between the parties. Figure 7 illustrates a model of a fair exchange protocol that uses on offline TTP (optimistic fair exchange) [112,113].

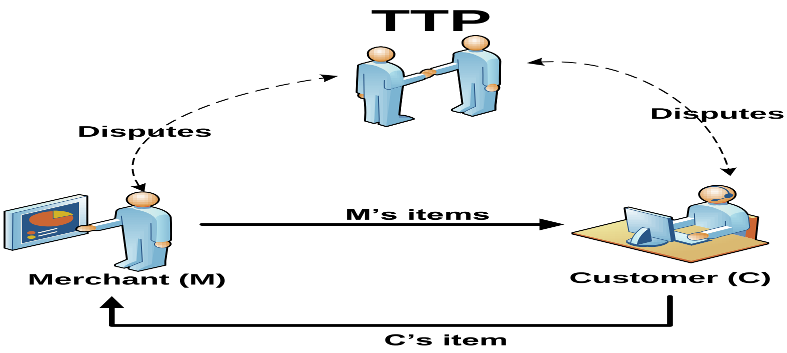


Figure 7: Offline TTP-based fair exchange model

In the optimistic fair exchange protocols, the role of the inline TTP is greatly decreased because of the minimal use of the TTP (it is not involved in every exchange) [112,113]. Examples of this kind of protocol are Appendix B.

• *3.2.3. Online protocols depend on a trusted third party.*

Such as in [43, 44, 45 and 46], these involve less participation on the part of the TTP. In such protocols, the TTP is not used during the protocol run for delivering the parties' items; rather, it is for verifying an item, and generating and/or storing proof of exchange of items [38]. The example below illustrates the use of an online TTP in these fair exchange protocols.

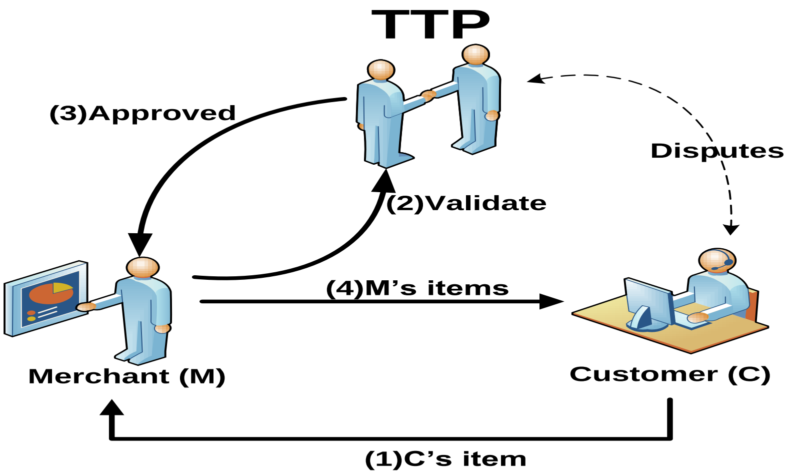


Figure 8: Online TTP-based fair exchange model

If the commodities to be traded between the transacting parties are a digital product and a payment, the customer starts the exchange, and when the payment is received by the merchant from the customer, the merchant then verifies it with the TTP (a bank for example) before sending the digital product to the customer. The TTP must therefore be online for the exchange process to be completed, and should be contacted in case there is any dispute. Figure 8, showing a model of a fair exchange protocol that is based on an online TTP, reveal that there is minimal involvement on the part of the TTP in this type of protocol, but the TTP must be available throughout the exchange process. This can be viewed as a drawback because the TTP may become the source of a communication bottleneck. In addition, the TTP might be targeted by dishonest users.

- ***The Zhang et al. Protocol***

Zhang *et al*. [43] suggested a fair exchange protocol that uses an online TTP. This protocol is for the exchange of items such as a physical product and a payment.

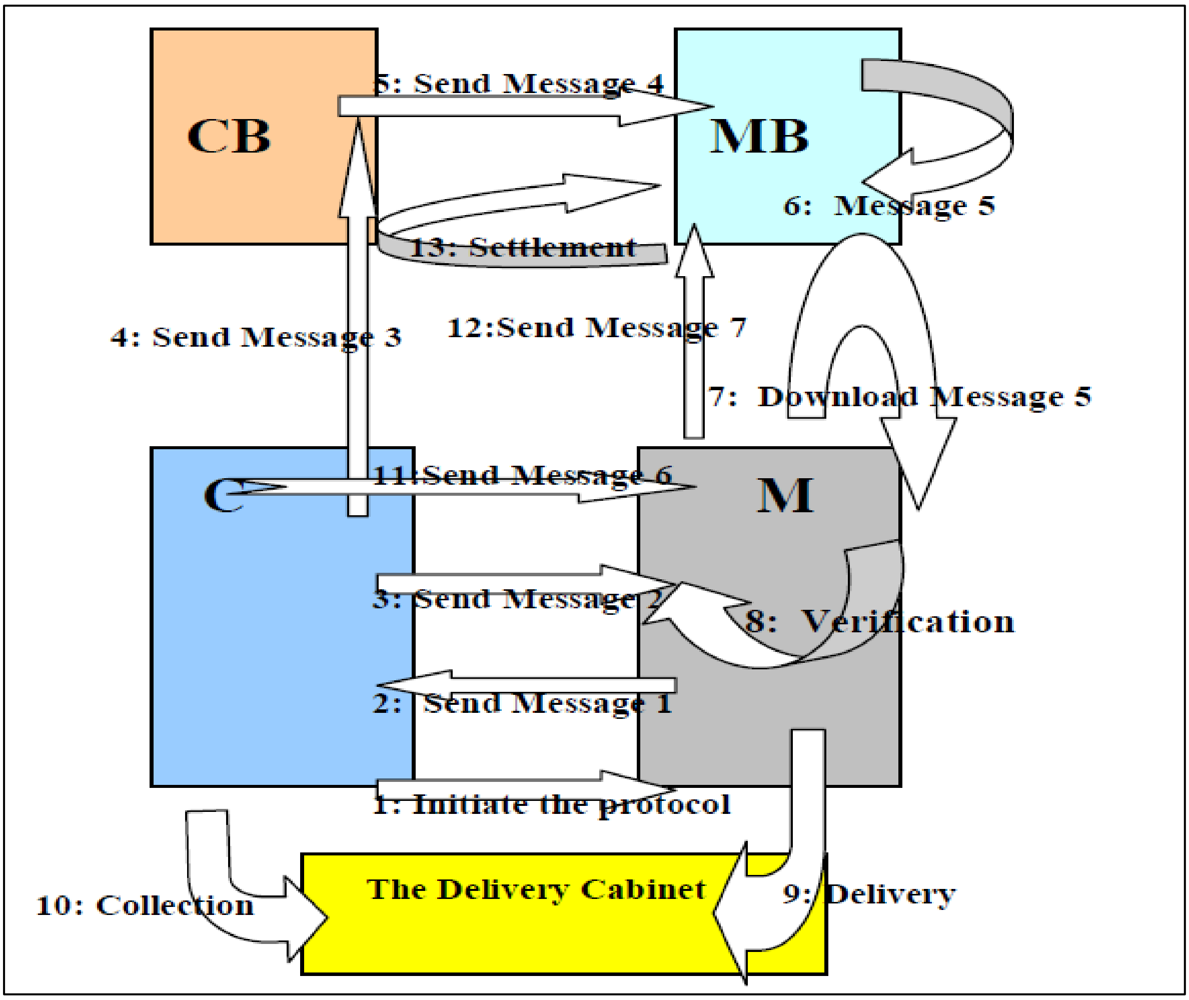


Figure 9: The process of Zhang *et al*.'s protocol [43]

The customer makes online payment (i.e. via the protocol messages) to the merchant, whereas a delivery agent is used to deliver the product to the customer, which means that the product is not transmitted electronically. The protocol is based on the theory of cross-validation [76]. In this protocol, the customer first begins the process by ordering a product from the merchant. The merchant then sends the invoice to the customer. Once the customer is happy with the invoice, they then firstly send a coded payment to the merchant and secondly to the TTP (the bank).

It is taken for granted that the merchant can download the coded payment (that was sent by the customer to the TTP) from the TTP (the bank). The merchant then makes a comparison of the two encrypted payments (i.e. the one received from the customer and the one downloaded from the TTP). If the merchant is satisfied that the encrypted messages compare, it means that the payment is valid. The merchant then delivers the product to the delivery agent after confirming the coded payment. The customer then takes the product from the delivery agent and, after confirming that the right product has been sent, they send the decryption key to the merchant who will then decode the coded payment [112,113].

We observe a limitation in the fairness of this protocol. If the merchant claims that he received an incorrect decryption key for the payment token or did not receive one at all, the third party (bank) will provide the K_1^{-1} (decrypts the contents of the payment) after asking the customer if he is satisfied. The third party (bank) will also provide the K_1^{-1} if the customer is not traceable. However, if the customer is not intentionally untraceable and also does not have the required product, then by having the K_1^{-1} from the third party, the merchant certainly has an advantage. The fairness of the protocol is based on the theory of cross-validation, which proceeds via a number of process steps that take into account the accuracy of the commodity. According to this protocol, the product information is not revealed to either the third party or the merchant in order to safeguard the customer's anonymity. A secured channel is proposed in this protocol to ensure confidentiality. Also, timeliness is a strong property in this protocol [112,113].

- ***The Devane et al. Protocol***

Devane *et al*. [44] suggested a fair exchange protocol that can be used for buying items online. This protocol enforces the fair exchange of a payment from a customer and a digital product from a merchant. However, in this protocol, a bank acts as an online TTP in which both the customer and the merchant have accounts [112,113].

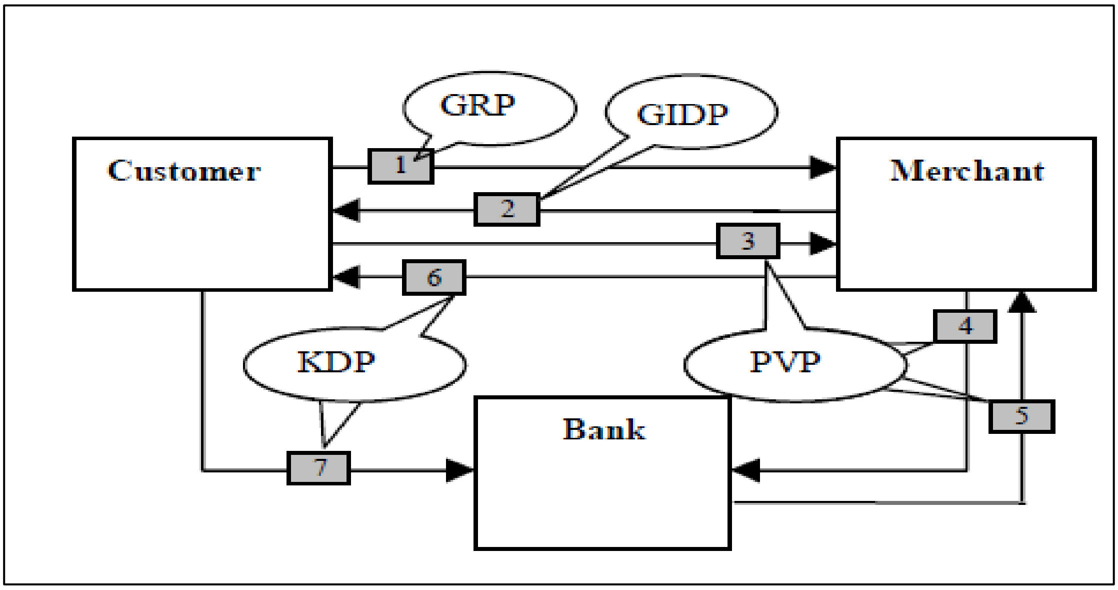


Figure 10: The process of the Devane *et al*. protocol [44.]

In the Devane *et al*. protocol, there are seven messages that are exchanged by the transacting parties and the TTP (which is the bank) during the exchange phase. The protocol begins by the customer sending the first message with a signed purchase request. Upon receiving the first message, the merchant authenticates it and, after approval, submits the second message with a signed invoice together with the encrypted digital product to the customer. After receiving the second message, the customer verifies and authenticates the signed invoice and, if satisfied, sends the third message to the merchant, which includes a signed payment.

After receiving the third message, the merchant verifies and authenticates it and, if satisfied, sends the fourth message to the bank with the decryption key for the digital product together with the third message that was sent by the customer and signed by the merchant (i.e. the merchant signs the signed payment by the customer and sends it to the bank). Upon receiving the fourth message, the bank authenticates it and, if it is approved, the bank then submits the fifth message to the merchant with the bank's signature on the signed payment together with the decryption key. After receiving the fifth message, the merchant then forwards it to the customer. After receiving the sixth message, the customer receives the decryption key and decrypts the encrypted digital product that was delivered in the second message. The customer submits the seventh message to the bank after approving that the decrypted digital product is the one that was described in the first message.

The seventh message contains the customer's approval of the digital product. Upon receiving the seventh message, the bank then finalizes the transaction by deducting the payment from the customer's account and transferring it to the merchant's account.

We observe a limitation in the fairness of this protocol; the merchant will receive the payment only after the customer has confirmed the items but there is no guarantee that the customer will make the payment after acquiring the items. The customer is certainly in an advantageous position. A secured channel is proposed to ensure confidentiality. Also, timeliness is a strong property in this protocol. The protocol takes into account the accuracy of the commodity but does not ensure the customer's anonymity [112,113].

- ***The Q. Zhang et al. Protocol***

Q. Zhang *et al.* [45] created a protocol that gives users a centric online m-payment solution. The protocol guarantees fair exchange and anonymity for the customer.

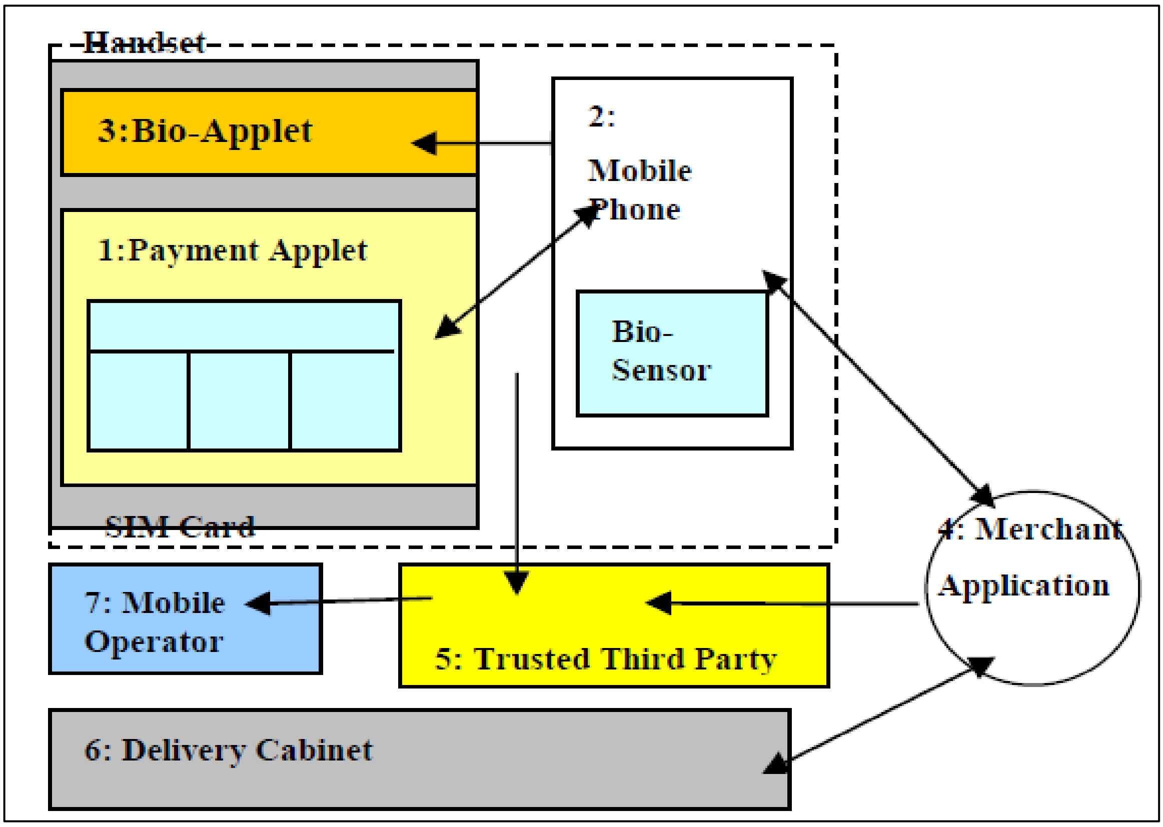


Figure 11: Principal participants in the Q. Zhang *et al*. protocol [45]

There are seven main entities involved in this protocol and there are twelve messages exchanged in this protocol with the participation of a trusted third party. And it is depending on the following assumptions: to begin with, the customer buys a pre-paid SIM card from a mobile phone operator without exposing the personal details. The mobile operator has a Commit buffer along the buyer's invoice account, and keeps a currency account with an authorized dealer. A PK/ SK pair is then created by an online TTP. All data regarding delivery addressees are saved by the customer in the cell phone. A fingerprint sensor is embedded within the device, and the customer then saves the fingerprint data inside the SIM card. In order to exchange data, the mobile phone is linked to the payment Applet through a shared symmetric encryption session key.

The customer initiates the protocol by accessing the e-commerce website to choose a commodity, and then he/she starts the process of transaction by signing in, using a mobile phone number. He/she then fill in the required data in the invoice and post it to the merchant. The data to be filled in include: the total amount payable, the identity of the merchant, the agreed price of the commodity, the quantity, the product identity, and the public key of the merchant. When the mobile phone receives the invoice from the merchant, the biometric data is extracted and transmitted to the Bio-Applet in the SIM card in order to establish the authenticity of the owner of the device.

Using Message 2 and the subsequent messages, the Bio-Applet then transmits the matching results to the mobile phone. The messages exchanged between the transacting parties are encoded with a session key to save the private details of the parties. If the results do not match, then the mobile phone operator terminates the protocol. On the other hand, if the results match, the mobile phone operator posts the payment request, the invoice, the seller's PK and the TTP's PK (encoded with a tsk) to the payment Applet. When the payment Applet receives the above requests, it posts the payment the buy command signed by the consumer and the consumer's PK. All the data are encoded with the seller's PK for security reasons. The mobile phone then transmits the same message to the merchant through Message 6.

By using a mobile phone, the payment Applet posts Messages 7 and 8, which contain the payment token (encoded by K, and K_1), K_2 and the PK of the payment Applet (encoded by the TTP's public key). After the trusted third party receives M8, it decodes it in order to extract the payment token, and then posts it to the mobile phone operator. The mobile operator then approves the transfer of the payment from the buyer's account to the Commit buffer. In case the insufficient credit in the customer's account, the mobile operator requests the customer to 'top up'. If the payment is successfully deducted from the buyer's account, the payment is transferred into Commit buffer.

In the final stage, the merchant posts (to the mobile phone) evidence of approval and terms and conditions regarding the process of the transaction through a signed buy requst, which is countersigned by the payment Applet. The seller then sends the commodity to the customer. The payment Applet then transmits [K_2, MApub] in order to decode the payment token to the seller. The seller lastly sends the payment token to the mobile phone operator. Financial institutions such as banks are not involved in this protocol. For protect the safety and personality of the parties of the transaction, the protocol requires all sensitive information to be saved in the SIM card. The SIM card runs the whole system of creating keys, encoding, decoding and issuance. The protocol guarantees the safety and privacy of the participants by using secure channels.

It also uses digital signatures through the senders' private keys in order to ensure that no party disowns a request or any feedback. Issues regarding the anonymity of the customer are satisfactorily addressed because the customer's personal data are not registered when he/she purchases the card. Neither the mobile phone operator nor the TTP are aware of the product information.

According to the designers of the protocol, problems that arise during the exchange are automatically resolved without the need for manual intervention, hence ensuring fairness for all parties. The protocol also has a mechanism in place that prevents any false allegation from either party. For example Message 10 has to be signed by the MA(*The Merchant's Application*)in order to counter any false claims by the merchant that he/she did not accept the terms and conditions. On the other hand, the TTP tells the customer to offer proof and reveal the delivery cabinet history in case there is the false allegation that he/she has not received the requested item.

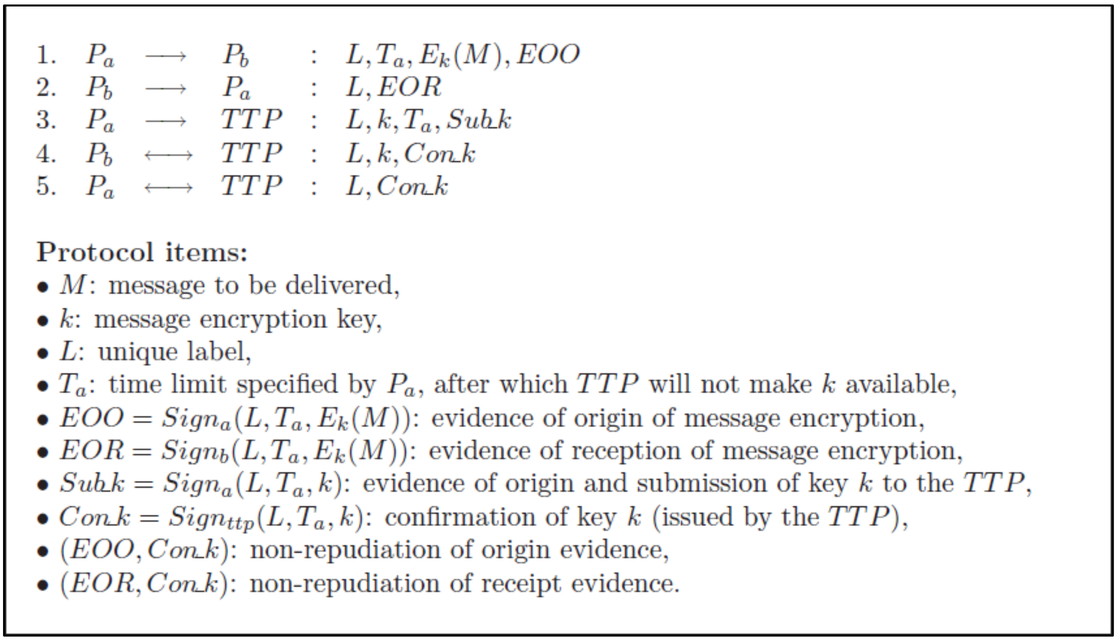
Although the protocol ensures fairness to a higher degree, there is still a weakness. For example, consider a scenario where the merchant alleges that he/she never received the decoding key for the payment token or received an incorrect one. In this situation, the TTP begins an extended protocol by first inquiring the consumer is happy with the product. If the consumer is happy, then the trusted third part transfer $K_1$to the merchant. Likewise, if the customer vanishes without a trace, the TTP will also give the K_1 to the merchant. Fairness will be become an issue if the customer has not acquired the item and cannot be traced due to unavoidable circumstance; then, the TTP provides the K_1 to the merchant. In this case, the merchant will be in a position of unfair advantage, hence creating a fairness issue.

We can see that this protocol has weaknesses in terms of fairness. This issue of fairness can arise when the merchant appeal aslo the consumer able falsely claim that the seller has not posted the items; the TTP must then request the seller to present the the history of delivery cabinet and submit proof.

There are other limitations in terms of the fairness of this protocol. For example, if the merchant makes an allegation that he/she received an incorrect decoding key for the payment token (or never received it), then, according to the extended protocol, the TTP will issue the K_1 after gaining the customer's consent. If the customer is untraceable, the TTP will also issue the K_1 to the merchant. Consider a scenario where the customer is not traceable due to unavoidable circumstance and also has not yet received the requested item, and then by the acquiring K_1 from the TTP, the merchant will definitely be in a position of unfair advantage.

- ***The Zhou and Gollmann Protocol***

Zhou and Gollmann proposed a non-repudiation protocol that uses an online TTP [46]. The objective of this protocol is to minimize the role of the TTP, to provide the originator and the intended recipient with evidence both during and after the execution of the protocol (without any party having an unfair advantage), and to divide the message M into two parts: a commitment C and a key k. The commitment is transmitted from the originator A to the recipient B, and then the key is lodged with the TTP. Both parties must acquire the confirmed key from the TTP as part of the non-repudiation proof needed in resolving a dispute [112,113].



$$\begin{array}{llcll} 1. & P_a & \longrightarrow & P_b & : L, T_a, E_k(M), EOO \\ 2. & P_b & \longrightarrow & P_a & : L, EOR \\ 3. & P_a & \longrightarrow & TTP & : L, k, T_a, Sub\_k \\ 4. & P_b & \longleftrightarrow & TTP & : L, k, Con\_k \\ 5. & P_a & \longleftrightarrow & TTP & : L, Con\_k \end{array}$$

Protocol items:

- M: message to be delivered,
- k: message encryption key,
- L: unique label,
- T_a: time limit specified by P_a, after which TTP will not make k available,
- $EOO = Sign_a(L, T_a, E_k(M))$: evidence of origin of message encryption,
- $EOR = Sign_b(L, T_a, E_k(M))$: evidence of reception of message encryption,
- $Sub\_k = Sign_a(L, T_a, k)$: evidence of origin and submission of key k to the TTP,
- $Con\_k = Sign_{ttp}(L, T_a, k)$: confirmation of key k (issued by the TTP),
- (EOO, Con_k): non-repudiation of origin evidence,
- (EOR, Con_k): non-repudiation of receipt evidence.

Figure 12: Zhou and Gollmann's protocol [46]

During the protocol, if an invalid message is received or if an awaited message does not arrive, the prospective recipient terminates the protocol. In such a case, the following proofs are revealed: proof of origin, evidence of delivery, evidence of submission, and evidence of confirmation. The protocol is outlined as follows: the message to be sent consists of two parts, where one is the encrypted text C and the other is the key k. The sender A sends his digital signature and the encrypted text C to the intended recipient B.

A begins the protocol by transmitting the cipher, using the session key k of the message he needs to transmit to B, a tag that marks the protocol session, a time-out value before which the session key should be sent to the TTP (and after which it can be consulted), as well as the signed non-repudiation of origin evidence for the ciphered message. A suggests a consultation time-out and if B agrees, he transmits his signed non-repudiation of receipt evidence for the ciphered message. After receiving it, A then transmits to the TTP a signed copy of the session key. The TTP only receives one submission from a party in the course of a protocol session. The TTP then confirms the validity of A’s signature and whether the time-out is exceeded or not. After the time-out, B may obtain the session key and the non-repudiation of origin evidence for this session key issued by the TTP. This evidence is required when making a full non-repudiation of origin evidence for the message that A submits to him. Similarly, A completes the non-repudiation of receipt evidence for the message by consulting with the TTP. The two parties, A and B, will then request the session key and the related evidence for this key from the TTP.

For B, the proof or evidence is an indication of origin, and for A, the evidence proves that B can access the key. Both parties can access, at the right time, a read-only public directory controlled by the TTP. If the gathered evidence cannot be obtained by one party, that party will lose any potential dispute on the issue. In this case, the role of the TTP is minimized by obviating the obligation to acquire the data (controlled by the TTP) on the parties. A resilient communication channel between the TTP and the parties is necessary for the proper functioning of the protocol. If the channels of communication between the TTP and respectively A and B are resilient, the protocol is fairly strong and upholds the timeliness feature.

In a fair non-repudiation protocol, the execution of the protocol should ensure that the Non-Repudiation of Delivery Token (NRDT) and the Non-Repudiation of Origin Token (NROT) are accessible to both the originator and the intended recipient, respectively. In addition, the protocol should be fail-safe. In other words, an incomplete execution of the protocol will not lead to a scenario where the NRDT is accessible to the originator but the NROT is not accessible to the intended recipient, or vice versa.

According to the definition of fairness, the protocol is not fair. This is because if B gives up after B finishes the first step, B does not know the subject matter of the message, but he receives the Non-Repudiation of Delivery Token. Besides, the protocol is designed to transport more messages when running and it includes C in the evidence, which increases the amount of data transport. The correctness of the product property is not considered in this protocol. Also, the protocol does not ensure the customer's anonymity, as it starts with a message sent by A, which discloses the customer's true identity. A secured channel is proposed to ensure confidentiality; also, timeliness is a strong property in this protocol [112,113]. Summary of the critical review is shown in table 1.

Protocol	**Items to be exchanged**	**Fairness**	**Weak-nesses**	**Load on TTP**	**Efficiency**	**# messages (exchange phase)**	**Type of fairness**
Zhang *et al.*	Payment and a Product (digital or physical)	Yes	1	High	Medium	7 + physical delivery and collection	Strong
Devane *et al.*	Payment and digital product	Yes	1	High	Medium	7	Strong
Zhou *et al.*	Provide the originator and the intended recipient with evidence after an execution	Yes	3	High	Medium	5	Fairness is not ensured
Q. Zhang *et al.*	Payment and digital product	Yes	1	High	Medium	12	Strong

Table 1: Comparison of fair exchange protocols [4].

3.2.4. Protocols do not use a trusted third party

In these, the two parties involved in the transaction exchange their items without the involvement of a TTP. The gradual exchange protocols [47, 48] can be used when the commodities to be exchanged can be equitably partitioned into a number of parts. A gradual exchange protocol is based on the principle of having several rounds to complete the process of exchanging items between the transacting parties [112,113].

The parties each exchange an item in every round and the number of rounds is equivalent to the number of parts into which the commodities are divided. The process of exchanging commodities continues until the transaction is completed and each party has received what it expects. In each round, both the customer and the merchant send part of their commodity and also receive part of the other party's commodity (see Figure 14).The number of parts delivered to each party is almost the same at any given time [38].

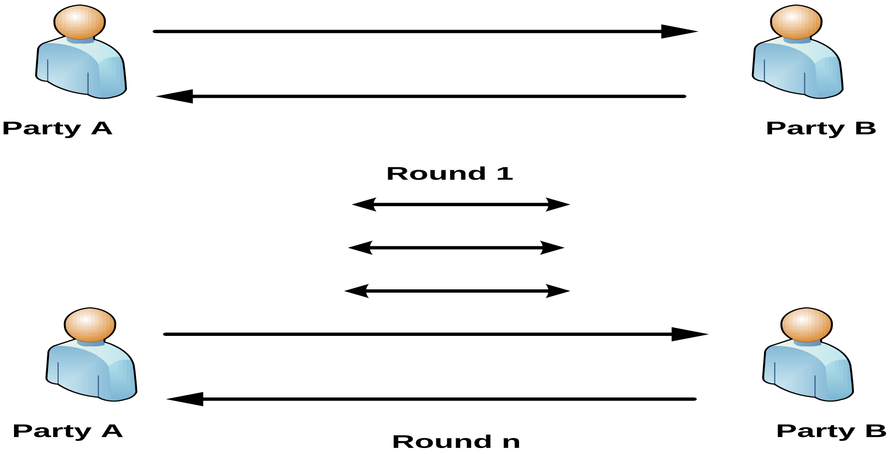


Figure 13: Gradual exchange protocols.

The major drawback of the gradual exchange protocols is that several rounds are needed to complete the exchange process. If there many rounds to be made, a number of communication steps are required, which can overload the communication channel used by the two parties? It is actually taken for granted that the items to be traded between the transacting parties have the same size [32]. As a result, this type of protocol does not support items of different sizes. Gradual exchange protocols lack the involvement of a TTP, which can be problematic, as it is impossible to guarantee fairness for both parties without a TTP who can mediate and solve problems [49].

The following scenario explains the reason why gradual exchange protocols do not ensure fairness. In stage one of the exchanges; side X submits the first part of his product to M. In stage two, M submits his items to X. Thus, the process of exchanging parts of the items continues. If X delivers its last part of the item to M, but M vanishes without sending the final part to X, then M will have the complete items of X while X still wait the last part of items of M. Therefore, there is a likelihood that one of the parties will suffer from an unfair dealing during the fair exchange protocol [49]. The following is an explanation of parties interchange money for a digital item and no use of the trusted third party [50]. A small amount of payment is sent by a buyer to a seller, and in return the buyer receives a small part of the digital product from the merchant. This process of exchanging items continuous until each party receives all the items. This means that the merchant receives all of the payment while the customer receives the complete digital product. In this scenario, there is the likelihood that after receiving the final part of the payment, the merchant may not honour the commitment of delivering the final part of the digital product. Therefore, fairness has not been achieved. Micro-payment systems [50, 51] can be used to carry out that kind of exchange [112,113].

- ***Comparison of fair exchange protocols***

Protocol	**Items to be exchanged**	**Use TTP**	**TTP Type**	**Load on TTP**	**Efficiency**	**# messages (exchange phase)**	**Type of fairness**
Nenadic *et al*	e-mail and receipt	Yes	Offline	Low	High	4	strong
Nenadic *et al*	Digital product and receipt	Yes	Offline	Low	High	4	Strong
Asokan *et al*	Signatures on a contract	Yes	Offline	Low	High	4	Strong
Asokan *et al*	Payment and digital product	Yes	Offline	Low	High	4	Weak or Strong
Jakobsson	Payment and digital product	No	N/A	N/A	High	Not specified	Fairness is not ensured
Burk and Pfitzmann	Payment and digital product	Yes	Inline	Very high	Low	7	Strong
Zhang *et al*	Payment and a product (digital or physical)	Yes	Online	High	Medium	7 + physical delivery and collection	Strong
Ray *et al*	Payment and digital product	Yes	Offline	Low	High	4	Strong

Zhang *et al*	Two digital documents	Yes	Offline	Low	High	4	Strong
Devane *et al*	Payment and digital product	Yes	Online	High	Medium	7	Strong
Ray *et al*	Payment and digital product	Yes	Inline	Very high	Low	6	Strong
Alaraj *et al*	Payment and digital product	Yes	Offline	Low	High	3	Strong

Table 2: Comparison of fair exchange protocols [4]

3.3. Dispute resolution

Sometimes disputes arise between the parties involved in a transaction. For example, a customer may receive from a merchant a product that is different from the one they expected (or the product may have a fault). There is a high probability of a dispute occurring in an online purchase between the seller and the buyer, as they do not trust each other and the customer is not in a position to physically see and try the product.

Dispute resolution formally occurs in a court of law. However, there are some options for dispute resolution without resorting to a court of law, and they are known as Alternative Dispute Resolution (ADR); mediation and arbitration are models of Alternative Dispute Resolution.Online dispute resolutions are crucial in e-commerce as the merchants and customers may be in different geographical locations and hence operating under different legal systems. When a disagreement is resolved online, it is define as Online Dispute Resolution (ODR), and there are many methods used in ODR [52, 53]:

Adjudication

Adjudication refers to presence indifferent TP who gathers data from the parties involved in a dispute. After gathering the evidence, the TP then takes a binding decision.

Evaluation

Evaluation is similar to adjudication but the decision reached by the indifferent TP is not binding, rather a recommendation.

Mediocrity

Mediocrity refers to having a TP who assists the parties involved in a dispute to reach an agreement. So, it is different from adjudication where the TP makes the binding decision.

Auto parley

Auto parley for solve disputes that are concerned with pecuniary issues. Depend on the principle of Presence blind tenders from the parties, who offer their propositions to solve the problem. Each party's offer is hidden from the other and at last, software suggests a settlement if the propose of the parties are almost the same.

Pseudo judgment

A pseudo judgment is depending on Presence jurors who offer their services voluntarily and who make a decision on a disputed case using an online site platform.The conclusion took by the jurors is not bound.

Claim support

Claim support is a device that assists the individual who initiates a dispute. Some of these devices include interacting format that the person must finished. These devices provide counsel and detail like situation that have been helped by the device.

Credit Card Charge Back

In this technique, the credit card issuer’s act as a TP between the two parties involved in a transaction. The credit card issuer assesses the dispute and if the buyer’s complaint is acceptable, it issues the buyer’s value back.

Although the above techniques are done online, they are not fully automated, and hence they have to be performed manually.

There are two kinds of fair exchange protocol that contain dispute resolution [16]. The first type offers the dispute resolution mechanism during the execution phase, while the second type does not contain any dispute resolution steps during the execution. Protocols that contain dispute resolution during the execution phase have a mechanism that guarantees a resolution should there be a problem. In these protocols, a TTP receives a complaint from the disadvantaged party and resolves the dispute fairly. Some examples of this kind of protocol are [33, 35, 40, 41, 42, 43 and 44].

Protocols that do not have a dispute resolution during the execution phase are based on gathering strong evidence to be used in the resolution of a problem. These proofs are presented after the fair exchange protocol is executed, and the resolution can be made through an ODR mechanism or in a court of law. The gathered evidence will help in resolving a dispute between the parties; examples of this type are [54, 55].

In order to use an ODR technique in fair exchange protocols, the protocol should offer a means of storing the evidence so that it can then be used in the ODR technique. This method of resolving problems (using ODR techniques) is called an 'after the fact' solution [35], in which the transacting parties may vanish. In this case, the dispute resolution should be part of the execution of the fair exchange protocol, not after the execution of the protocol has been completed. In order to achieve this, there should be a means through which disputes can be solved without interference (with the use of the trusted third party) before the execution of the protocol has been completed.

There is no doubt that the protocols that have dispute resolution over the enforcement phase of the protocol are better than the protocols that do not; this is because the transacting parties are often in different geographical locations, and it will be difficult to go to a court of law in the other party's country. In addition, one party may vanish without trace before the process of exchange is completed. Therefore, the first type of protocols (those that involve dispute resolution in the execution phase) is better suited to e-commerce, as it is difficult to resolve a dispute after the execution of the fair exchange protocol.

The need for a dispute resolution mechanism can be reduced by finding a way of avoiding the possibility of a dispute arising in the first place. This means that before the parties exchange their items, every part should check that the product they will get is the correct one. If the possibility of a dispute arising is reduced, the number of messages required for the resolution of the dispute will also be reduced.

3.4. Protocol Design

E-commerce systems usually use electronic payment systems. The effective and successful implementation of these electronic systems is mainly reliant on the protocols used. In turn, the reliability of the protocols, in terms of being secure and sound, depends upon their design. A protocol can be defined as the rules, formats and procedures that have been mutually accepted by the transacting parties. The protocol specifies the procedures used for:

- Start and abortion of data reciprocity
- Coincidence of transmitter and remittee
- Exposure and rectification of sending fault
- Forms and encryption of data

A standard protocol description is composed of five different kinds. For the protocol to be full, each description should containt the following:

1. The kind of serving the protocol will render

2. The hypothesis about the implementation case of the protocol

3. The messages format applied to execute the protocol

4. The encryption of every message in the lexicon

5. Principles of the procedures that protect the uniformity of message exchanges [56]

3.5. Characteristics of a Protocol

Generally, all protocols have certain properties, and these can be summarised as follows:

- **Simplicity:** A good-formulated protocol is built around simple individual pieces, which are easy to understand and execute. Such protocols are also highly provable and sustainable.
- **Modularity:** Each small piece in a protocol should be light-weight and capable of being individually developed, authenticated, executed, and sustained. Perpendicular tasks are not combined; they are formulated as separate entities. For example, error and flow control are orthogonal functions and hence should be separate modules.
- **Protocol should be good formative:** A good-structured protocol should not be more than the particular and should not include any unachievable or un-executable code. In addition, a good-structured protocol should not be incomplete or under-specified. Also, a well-structured protocol is restricted; it should not spate the familiar framework determines, like the determine ability of message lines. Finally, a good-structured protocol is self-balance; this means that it should be fault-tolerant. These guarantee that the system is stable and can operate normally even when a passing fault uncharted variation the condition of

the protocol. The correct state is achieved after a finite number of execution steps.

- **Robustness:** A protocol should be prepared like method that it can operate in the full variety of networking conditions to which it may be exposed. It should be able to deal with unexpected challenges such as abnormalities in input.
- **Consistency:** Protocols have some standard failure modes. Three of the more prominent methods of failure are:

  Deadlock: this is a condition in which no additional protocol execution is possible.

  Lovelock: this occurs when commands are sequentially executed indefinitely without achieving any successful progress.

  Improper termination: this occurs when the implementation of the protocol is completed without fulfilling the appropriate termination conditions [56].

3.6. Rules of Design

In order to achieve the characteristics of the protocols discussed above, Gerard J. Holzmann in [57] has suggested ten basic rules for formulating a protocol:

1. Ensure that the issue is good formulated. Each standard for designing needs and restrictions should be determined before a layout is initiated.
2. Specify the kind of service to be carried out at each stage of the abstraction before choosing the designs.

3. Make the protocol simple. It is very difficult to authenticate and execute complicated protocols; also, they are generally inefficient. There are few complicated problems when formulating a protocol and the role of the designer is to identify and separate the easier problems after that fix the problem one by one.
4. Before executing the design of a protocol, create a high-level model and ensure that all the requirements of the design are fulfilled.
5. Execute the design, evaluate its capabilities, and if necessary try to modify it in order to achieve maximum efficiency.
6. Ensure that the last execution is the same as the high-level design that was authenticated.
7. Do not jump Rules 1 to 7.

Rule number 10 is often violated.

3.7. Cryptographic Concepts

The protocols mentioned in this thesis rely on cryptography as the underlying security model. This thesis is not about cryptography but this section has been included as a certain amount of background is required.

- ***Symmetric Encryption***

Symmetric encryption is the easiest and most explicit method of sending messages between a sender (X) and a recipient (Y). The two entities, X and Y, initially agree upon a combination number. X writes the message, places it in a safe box, locks it up and submits it to Y. When Y receives the box, he or she decrypts the message using the combination key that was agreed upon. On the other hand, Y can also transmit a message using the same method or ‘protocol’ that was used by X. We realise that this protocol has symmetry as both entities (X and Y) use an odd key for coding and decoding. Because of this feature, this type of protocol is known as symmetric encryption. If E stands for the encoding algorithm that is used, which relies upon a key K to encode plaintext P, then the equation for the cipher text C can be written as follows:

$$C = E_K (P)$$

The following equation is for the decryption step:

$$P = E_K(C)$$

We can also encode several times using separate keys, such as K_1, K_2, etc., with the same encoding algorithm in order to create a double encoded cipher text:

Or a cipher text that is encoded three times:

When decoding, the same keys are used in the opposite order to which they were originally used.

Symmetric encryption systems are extremely fast and are easier to implement than other systems that use separate protocols, such as asymmetric encryption. The main drawback of a symmetric encryption system is that it is hard to implement because of its requirements and methods concerning the key exchange.

Such requirements necessitate that all participants be configured with the secret key. Symmetric systems are widely used in a variety of fields, such as in financial institutions (principally banks) and in some military applications. Some examples of symmetric encryption systems are the Digital Encryption Standard (DES) and Digital Encryption Standard with triple encryption (DES3) and the Advanced Encryption Standard (AES) [58].

- ***Asymmetric Encryption***

In asymmetric encryption systems, the parties do not agree on a decryption key. For example, X encrypts a message with a secret combination key only known to him. If Y wants to transmit X a message, he or she must order an open lock from X. When Y obtains the open lock, he then writes his message, places it in a box, locks it up and transmits it to X. When X receives the box, he unlocks it and decrypts the message using the secret key only known to him.

We realize in this exchange that Y does not have to know the combination key that is known only to X but instead he can obtain the open lock from X when required. The same process can also be initiated by Y in order to obtain a message from X. It is evident that the procedures carried out by the two parties (X and Y) in sending and obtaining a single message is different. In this case we can say that the protocol is asymmetric; any encryption system that implements this protocol is known as asymmetric. Note that X could use this protocol to obtain messages from various senders as long as they can obtain access to one of his open locks. This can be achieved by X disseminating as many such locks as needed. In asymmetric encryption, both parties should have two keys; a public key that is shared openly by both parties, and a private key that is kept secret. When implementing asymmetric cryptography, the message is encoded using the meant remittee's PK. The message is encrypted by the remittee using the SK. Contrary to symmetric encryption, asymmetric code face to be calculational intense because they need lengthy keys, hence when using them in public key cryptography they are normally used in association with symmetric systems. Asymmetric encryption is often used to transfer a session key rather than information proper plaintext. This session key is then used to encode information using a symmetric encryption system. Therefore, this provides the key exchange advantages of asymmetric encryption in combination with the speed of symmetric encryption. The extreme usually used asymmetric code is the RSA algorithm, which is also referred to as public key cryptography. This algorithm is depending on the use of specific prime numbers, through which the private and PK are generated for achieve the protocol. These prime numbers are generated by a third party whose role is mainly to distribute the public and private key pairs.

This method or infrastructure for generating prime numbers is referred to as the Public Key Infrastructure or PKI. When communicating with a number of people, the use of a public key is suitable because it helps us avoid multiple key-exchanges. The public key provides the basis for cryptanalysis:

$$\boldsymbol{C = E_K (P)}$$

Where K refers to the public key, P can be predicted by the analyst who then confirms the answer by comparing C with the intercepted cipher text. The prediction will be simple if it is based on data that are thought to be an element of the plaintext. Hence, public key algorithms are usually formulated to avoid plaintext attack.

Studies on public key and asymmetric systems have shown that their level of security is not as high as the one that can be attained with a well-formulated symmetric system [58].

- ***Public-Private Key Encryption***

Public-Private Key Encryption [59, 60] is basically asymmetric, and when it comes to the box and combination-lock paradigm, it is based on considering a lock that has two combinations: one to open the lock and the other to lock it. A second level of security is a fundamental feature, built on a basic assumption: combination locks can be locked regardless of the rotor positions. For instance, consider that Tom writes a message and then locks the box with a special lock formulated by Henry, using a combination number that is unique to Henry but is easily available to Tom and any other entity that wants to write a message to Henry. We can analyse the above exchange by saying that the combination number is the same as the public key. When Henry receives the message, he unlocks it using a combination number that is known only to himself, and this is the same as a private key. In order to create such a lock, there must be some mechanical ‘property’ linking the combination numbers needed to initially lock the box and open it later. This represents the main security feature of public and private key encryption. This is because this feature deals with some specific and precise relationships that are unique to the use of prime numbers and their applications, concerning the creation of pseudo-random number streams and stochastic functions in general [61].

- ***Hash Function***

The hash function refers to a function that takes a string of any length as an input, and returns a fixed length as an output [62]. The output of the hash function is defined as the hash value. The hash function is referred to as a one-way function if it is computationally impossible to obtain the original input from the hash value [62]. Let us take for example H to be the hash value, and it is computed for a message M. In this case, we cannot obtain M from the hash value H. The hash function is referred to as a strong-collision-resistance hash function [62] if it is computationally impossible to have the same hash values for separate messages. For example, if we have two separate messages, M1 and M2, and the hash values for the two messages are H1 and H2, respectively, then H1 ≠ H2.

- ***Digital signatures***

Digital signatures are used to implement electronic or handwritten signatures. The digital signature is a development of public key cryptography [63]. When creating a digital signature, a private key is required, and when authenticating the signature, a public key is required. We stated in the public key cryptography section that each entity has a SK and a PK that are mathematically connected to each other. For encode a message, the transmitter uses the PK of the recipient.

On the other hand, the recipient of the encoded message uses the PK for decodes the message. However, in a digital signature, the sender uses his own SK to digitally sign the message, and then transmits it to the recipient. The recipient of the message then uses the public key of the sender in order to authenticate the identity of the sender of the message. This is mainly due to the fact that the private key is secret and is known only by the owner, while the public key is accessible to all. Although there are various algorithms for digital signatures, in this thesis we will only discuss the RSA signature.

- ***RSA Signature***

RSA uses a public key to encrypt the message and a private key to sign it. In order to create the RSA signature [64], the hash value of the message to be signed is computed, and then the hash value is encoded with the private key of the person who signed the message. After that, the message together with the signature is sent to the recipient. The following function shows how the sender creates a signature for a message M:

$$\boldsymbol{Sig\ (M) = (h(M))^d\ (mod_n)}$$

To authenticate the signature [64], the recipient of the message and the signature uses the public key, pk, of the person who signed the message when decoding the signed hash value. After that, the recipient computes the hash value of the message and compares the hash value computed by the receiver against the one decoded using the public key of the signer. If the result matches, then the signature is authentic. The function below shows the operation for decoding the signature. In this case, H stands for the hash value of the signed message:

$$\boldsymbol{H = (Sig\ (M))^{e}\ (mod_n)}$$

3.8. Summary

In this chapter, a general description of electronic commerce has been presented (in the introduction), and the advantages of e-commerce and e-payment systems have been outlined. The concept behind fair exchange protocols has been described; these protocols are intended to ensure fairness for both parties involved in a transaction. Fairness in the context of the fair exchange protocols between the merchant and the customer is achieved in the transaction if, at the end of the protocol execution, each party involved in the exchange receives the item of the other party, or none do [112,113]. A number of fair exchange protocols have been reviewed in this chapter, which can be categorized into two types; the ones that do not involve the use of a TTP and the ones that do. The former allows the parties to exchange their items gradually bit by bit (in parts) until the items are fully exchanged in a fair manner. The latter is divided into three types. The first type involves the use of the TTP for delivering the items to the parties involved in the exchange; these are inline TTP-based fair exchange protocols. The second type uses an online TTP, where there is minimal involvement on the part of the TTP. The third type uses an offline TTP (optimistic fair exchange protocols). In the optimistic fair exchange protocols, the involvement of the TTP is only initiated if there is a problem during the protocol execution. Table 2 gives a summary of some examples of these protocols [112,113].

Resolution of disputes in the context of fair exchange protocols has been discussed and some of its techniques were presented. It has been clearly demonstrated that dispute resolution should be included in the fair exchange protocol to protect both parties from unfair dealings. Therefore, a fair exchange protocol that reduces disputes and includes automated dispute resolution is required. As can be seen in Table 2, there is minimal involvement on the art of the TTP in those protocols that use an offline TTP because the TTP will only be used if there is a problem.

In inline TTP-based protocols, the TTP is highly involved because the TTP receives the parties' items, verifies them, and if they are correct, the TTP then delivers them to the parties. Therefore, efficiency is low in protocols that are based on an inline TTP, as the TTP does everything, even in cases where the transacting parties are honest. The efficiency of protocols that are based on an online TTP is medium because the involvement of the TTP is relatively high, as the TTP will only be used during the exchange of items. This chapter also has covered the cryptographic concepts that will be used in this thesis [112,113].

Chapter 4
The proposed protocol

Objectives:

- *The proposed Protocol for digital and physical product*
- *Dispute resolution*
- *Dispute Analysis*
- *Analysis of Scenarios*
- *Comparisons of Protocols*
- *The Timing of the Protocol*
- *Cost functions for the proposed protocol*
- *Comparisons with Other Protocols*

This chapter presents the fairness protocol for purchasing items over the Internet. The protocol guarantees the security of the communication channel and fairness to the customer and merchant. The security of the communication channel is one of the main factors in fairness. Two main protocols will be discussed in this chapter. The customer contacts the merchant and makes a purchase over the Internet. The product can be either sent to the customer via the Internet or by post. Therefore we discuss two different protocols on how to ensure fairness when using the Internet to deliver the item, and how to ensure fairness when using the post to deliver the item.

• *Assumptions*

The hypothesis below are used in the proposed protocol:

1. All entities use similar algorithms when encoding, decoding and signing.
2. The customer opens an account with a FSP and Merchant registers with FSP.
3. The FSP keeps a general index server, to connected to the seller's account. The seller has permission to enter to the site and download messages from it.
4. All encryptions should be secure enough so that the recipient of the message cannot decode it without the proper key.
5. The customer processes the payment through the FSP.
6. Safe and secure channels are created between the entities during the exchange process.
7. The pre-exchange stage takes place after the Customer identifies the commodity he or she wants to buy. It is assumed at this stage that both the transacting parties are agreed on the item to be exchanged and the price.

8. The channels of communication between the FSP and the Customer, and between the FSP and the Merchant are resilient. A channel is said to be resilient if all submitted messages are obtained by their intended recipients. In the case of communication channel failure, fault tolerance techniques need to be applied to the proposed protocol.
9. All entities have faith and trust in the FSP in the sense that it will not act fraudulently or conspire with any entity.
10. The transacting parties agree on the FSP to be used for dispute resolution during the exchange and pre-exchange phases of the protocol.

- ***Notation***

The notations used in the description of the proposed protocol are summarized in Table 3:

Symbol	**Interpretation**
C, M, FSP	IDs for Customer, Merchant and Financial Service Provider
N	Invoice
D Di Pi	Product Product information Payment information
A → B : X	A sends X to B
X → Y	Transmission from entity X to entity Y
PK	Public Key
SK	Secret Key (private key)
TSK	Temporary Session Key
X:PK	Public Key of entity 'X'
X:SK	Secret Key of entity 'X'
X:PKS[ ]	The data are Signed using the Private Key of entity 'X'
X:SKE[ ]	The data are Encrypted using the Secret Key of entity 'X'

Table 3: Notations for proposed protocol

4.1. The proposed Protocol

- *For digital product*

This protocol discusses the communication channels and their communication contents when ordering an online product that can be delivered via the Internet. There are three main parties involved in this protocol. The Merchant (M) is the user with the product who would like to sell it, and the Customer (C) is the person who wants to buy the items from the seller. The Customer must pay the merchant either before or after receiving the product.

The payment over the Internet is managed by a Financial Services Provider (FSP). The Customer transfer the payment to the FSP, and it then transfer the payment to the Merchant. Thus, both the Customer and the Merchant should be registered with the FSP before invoking this protocol. The FSP is the entity that ensures the fairness of the protocol; it makes sure that the buyer gets the right item and that the seller gets the correct payment for that product. If the product is not transfer to the buyer by the seller, then the FSP takes the necessary steps to ensure fairness.

The following are the main entities in this protocol:

- ***Merchant:*** the entity that sells the product
- ***Customer:*** the entity that wants to buy the product
- ***Financial Service Provider:*** a third party that is trusted by the seller and the buyer. The FSP should be capable to process the payment (credit card, PayPal, etc) from the Customer and then credit the Merchant. Meanwhile, it should be able to reverse the transactions of the two parties should a dispute arise. When a dispute arises because a product has not been delivered to the Customer (after

payment), this party acts to resolve the dispute, such as taking the necessary steps to send the product to the Customer; this ensures fairness in the protocol.

- ***Protocol description***

- ***Pre-requisites***

The following are the pre-requisites of the protocol:

- The Merchant has outlet to the Internet and all the obtainable products are viewable on the Internet, or else the product information can be sent to another party via the Internet.
- The Customer has access to the Internet and he knows the identity of the merchant
- The Merchant registers with the Financial Service Provider. The customer should be aware of the presence of the FSP and its responsibilities. The Merchant should be able to accept payments from the FSP.
- The Customer should be able to submit payments through the FSP.

- ***Pre-exchange phase***

The Customer finds the Merchant's information on the Internet and views the list of products that can be purchased. Any product is sent directly to the Customer by the Merchant; it should not be altered by anyone in the middle. Thus, the integrity of the potential transaction is protected. Then, the Customer chooses the itemt hat he wants to buy locates the identification of the product and clicks the 'check out' button. The Merchant generates a temporary session key. This key can be used to code and decode the product; the security of this encryption is valid for a short period only (due to the small size of the key). The Merchant then encrypts the product using the temporary session key. The temporary session key will then be used by the buyer to decode the item. Meanwhile, the seller generates the invoice for the product. The invoice details the price that consumer should pay for the item. Then, the invoice and the temporary session key are sent to the FSP; the message cannot be sent as plain text. The merchant is now registered with the FSP.

The FSP has a public key certificate, and so the message is coded using the PK of the FSP. This encryption protects the confidentiality of the message. Before the message is encrypted, it is signed by the SK of the Merchant to protect the integrity of the message, so that no one else will be able to generate a fake message.

Pr-m1: Merchant → Financial Service Provider (FSP)

M: SK S [tSK, Invoice].

Message content: *Temporary Session Key, Invoice.*

The invoice consists of the next information:

- The product specifications, Di
- The identity of the customer, C
- The identity of the merchant, M.

The FSP verifies the invoice and then signs it using its SK, and transfer it back to the Merchant. Also, the message is coded using the public key of the Merchant. The digital signature on the invoice verifies that the invoice is received by the FSP, who has the relevant temporary session key, as the FSP is the only entity that has access to its private key.

The message is coded using the Merchant's PK to protect the confidentiality of the communication.

Pre-m2: Financial Service Provider (FSP) → Merchant

FSP: SK S [Invoice].

Message content: *Signed Invoice by the Financial Service Provider, Public Key Certificate of the Financial Service Provider.*

The Merchant receives the signed invoice from the FSP. Firstly, the message should be decrypted correctly through the private key of the Merchant, and then the invoice should be verified using the public key certificate of the FSP. If either fails, then the Merchant should be able to replay the message (pre-m1).

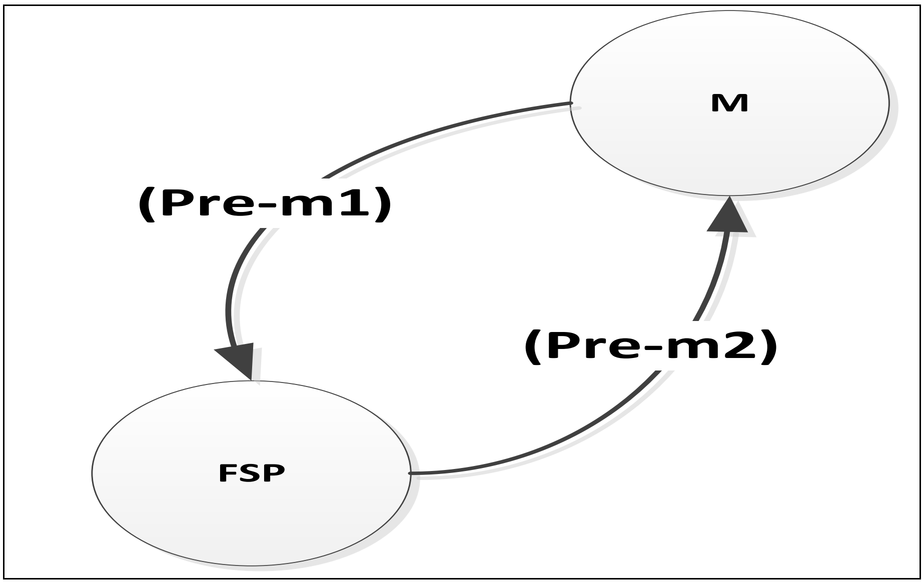


Figure 14: Pre-exchange phase

- ***Exchange phase***

Seller transfers the bill and the product to the consumer. The product is encrypted using the temporary session key. The whole message is signed using the SK of the Merchant; this is to protect the integrity of the message. Meanwhile, the message consists of the FSP's public key (or the Customer should be able to obtain this key from the Internet). The invoice is already signed by the FSP.

Meanwhile, the Merchant adds a timestamp onto the message in order to prevent 'replay' attacks, so that an attacker will not be able to replay the same message back to the customer later.

M1: Merchant → Customer

M: SK S [tSK E [Product], Invoice, FSP: PK, FSP: SK S [Invoice], Timestamp].

Message content: *Invoice, Encrypted Product, Information about Financial Service Provider, Timestamp.*

The consumer gets the message from the seller and extracts it. Firstly, he validates the digital signature on the invoice against the FSP's public key certificate. If the invoice is successfully validated, then the business between the Customer and the Merchant will be deemed fair (and will be guaranteed by the FSP). Then, the Customer finds the FSP's information in order to pay for the items. Meanwhile, the message has the item in it but the Customer will not be able to access it without the temporary key. Then, the consumer transfers the invoice and the payment details to the FSP. The message is coded using the PK of the FSP, and this key can be found in the public key certificate of the FSP. The payment information can be credit card details, bank information, etc.; this information will be visible to any other parties in this protocol.

M2: Customer → Financial Service Provider

C: PK E [Payment Information, Invoice].

Message content: *Invoice, Payment Information.*

The FSP verifies the invoice against the Merchant's public key certificate, and then processes the payment for the given amount. Then, he generates a payment confirmation. This payment confirmation and the sent invoice are signed using the SK of the FSP. This ensures the integrity of the message, and the message is then transfer back to the Customer. This is the declaration of that the consumer has made the payment for the issued invoice. Meanwhile, the payment confirmation can be found on the FSP's server, so that the Merchant will be able to find out if the Customer has indeed made the payment detailed in the issued invoice. If the seller is happy with the payment by the Customer, then he should release the temporary session key to the Customer. Only then will the Customer be able to decrypt the product and access it. So, the Merchant signs the temporary session key using his SK (to protect the integrity of the message), and sends the message to customer.

M3: Merchant → Customer

M: SK S [tSK].

The Customer decrypts the product using the tSK and starts using it. If the product cannot be accessed using the sent temporary session key, the Customer requests the merchant to resend the key. On the other hand, the Merchant is able to send the product directly to the Customer without encryption, as the payment has already been made.

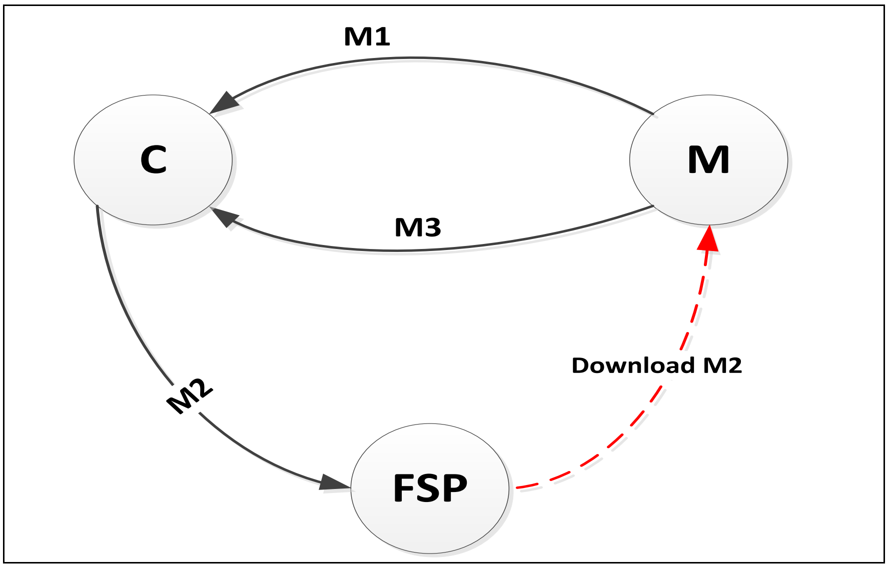


Figure 15: the exchange phase

- ***For a physical product***

- ***Pre-exchange phase***

The Customer finds the merchants information on the Internet and views the list of products that can be purchased. The Customer then chooses the item that he wants to buy, locates the identification of the product (product ID, product details and merchant ID) clicks the ‘check out’ button. The seller gets the product order from the consumer, and then generates an invoice for that product. The invoice states the price that the consumer should pay for the item. The seller sends the invoice to the FSP; this message cannot be send as plain text. The Merchant is now registered with the FSP. The FSP has a public key certificate, and so the message is coded using the PK of the FSP. This encryption protects the confidentiality of the message. Before the message is encrypted, it is signed by the SK of the Merchant to protect the integrity of the message (to stop someone else from being able to generate a fake message).

Pre-m1: Merchant → Financial Service Provider (FSP)

M: SK S [Invoice].

The invoice consists of the next information:

- The item specifications, Di
- The identity of the customer, C
- The identity of the merchant, M.

The FSP verifies the invoice and then signs it using its SK; it then transfers it back to the Merchant. Also, the message is coded using the PK of the Merchant. The digital signature on the invoice verifies that the invoice has been received by the FSP; it has the relevant temporary session key, as the FSP is the only entity that has access to its private key. The message is coded using the Merchant's PK to protect the confidentiality of the communication.

Pre-m2: Financial Service Provider (FSP) → Merchant

FSP: SK S [Invoice].

Message content: *Signed Invoice by the Financial Service Provider, Public Key Certificate of the Financial Service Provider.*

The Merchant receives the signed invoice from the FSP. Firstly, the message should be decrypted correctly through the private key of the Merchant, and then the invoice should be verified using the public key certificate of the FSP. If either fails, then the Merchant should be able to replay the message (pre-m1).

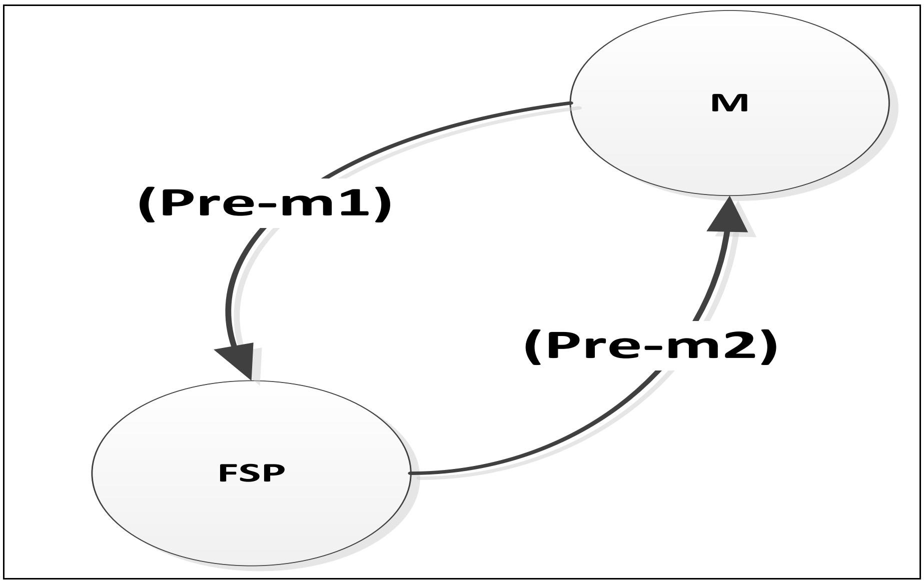


Figure 16: The pre-exchange phase of the Alternative Protocol.

- ***Exchange Phase***

The seller transfers the invoice to the consumer. The whole message is signed using the private key of the Merchant; this is to protect the integrity of the message. Meanwhile, the message consists of the FSP's public key (or the Customer should be able to obtain this key from the Internet). That public key contains information about the service provider to which the Customer should make the payment.

The invoice is already signed by the FSP. Meanwhile, the Merchant adds a timestamp to the message to prevent replay attacks, so that an attacker will not be able to replay the same message back to the customer later on.

M1: Merchant → Customer

M: SK S [Invoice, FSP: PK, R: SK S [Invoice], Timestamp]

Content of the Invoice: *Invoice: [Product ID, Merchant ID, Price]*

Message content: *Invoice, Information about Financial Service Provider, Timestamp.*

The consumer gets the message from the seller and extracts it. Then, the Customer validates the digital signature on the invoice against the FSP's public key certificate. If the invoice is successfully validated, then the business between the customer and the merchant will be deemed fair; it will be guaranteed by the FSP. The Customer will then find the FSP's information for pay the product.

To make the payment, the consumer transfers the invoice and the payment details to the FSP. The message is coded using the PK of the FSP, and this key can be found in the public key certificate of the FSP. The payment information can be credit card details, bank information, etc. This information will be visible to other parties in this protocol.

M2: Customer → Financial Service Provider

C: PK E [Payment Information, Invoice].

Message content: *Invoice, Payment Information.*

The FSP verifies the invoice against the Merchant's public key certificate, and then processes the payment for the given amount; the FSP then generates a payment confirmation. This payment confirmation and the sent invoice are signed using the SK of the FSP. This ensures the integrity of the message, and the message is then sent back to the Customer. This is the proof that consumer has pay the payment for the issued invoice. Meanwhile, the payment confirmation can be found on the FSP's server, so the Merchant will be able to find out if the Customer has indeed made the payment of the issued invoice. If the seller is happy with the payment by the customer, he then delivers the product to the Customer.

Merchant → Delivery Cabinet

Post the product.

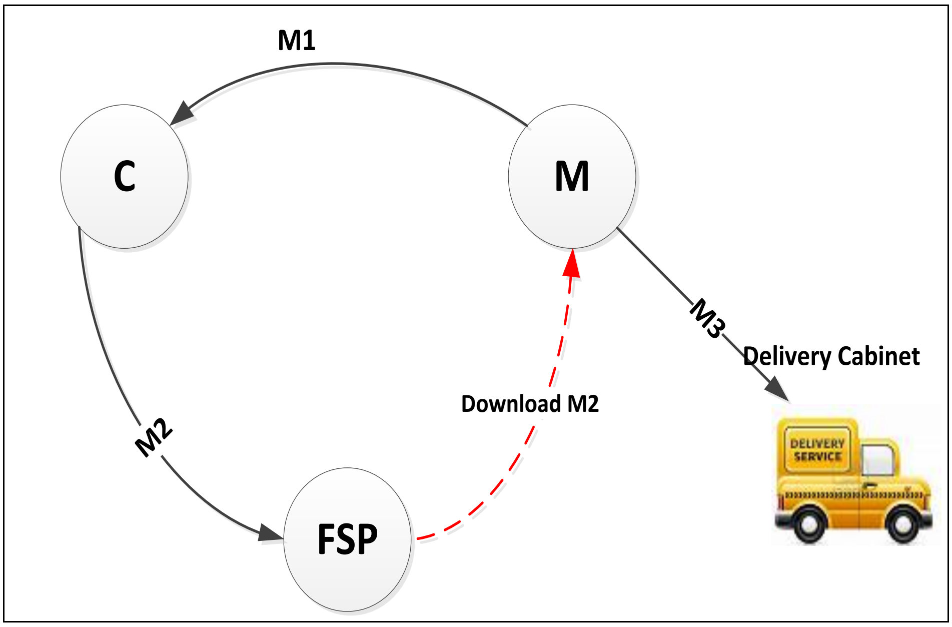


Figure 17: The exchange phase of the Alternative Protocol

The Customer waits for the product to be delivered, and then checks the product once it has arrived. If the product is not similar to the one ordered, then the consumer returns the item to the Merchant, who will then refund the money or send the correct product. When refunding, the Merchant contacts the FSP, who refunds the money back into the Customer's account.

4.2. Dispute resolution

There can be scenarios in which the temporary session key has not been sent to the consumer next he gets the payment conformation from the FSP. In that event, the consumer transfers the payment confirmation and the invoice to the FSP. This is the same message that was sent by the FSP after the payment was made.

Af-M1: Customer → Online Purchase Regulator

FSP: SK S [Payment Confirmation, Invoice]

Message content: *Invoice, Payment Confirmation.*

The FSP signs the invoice and the payment confirmation. The signed message is verified using the public key certificate of the FSP. Then the FSP verifies the digital signature on the invoice; it should be the signature that was inserted by it. If the digital signature is successfully validated, then the message is transfer to the seller, asking the seller to release the temporary session key to the Customer.

Af-M2: the Financial Service Provider → Merchant

FSP: SK S [Payment Confirmation, Invoice]

Message content: *Invoice, Payment Confirmation*

If merchant has not released the temporary session key, then the FSP has a copy of that key, which was given to it by the merchant when issuing the invoice. So, that key is sent to the Customer by the FSP.

Af-M1: Financial Service Provider → Customer

R: SK S [tsK

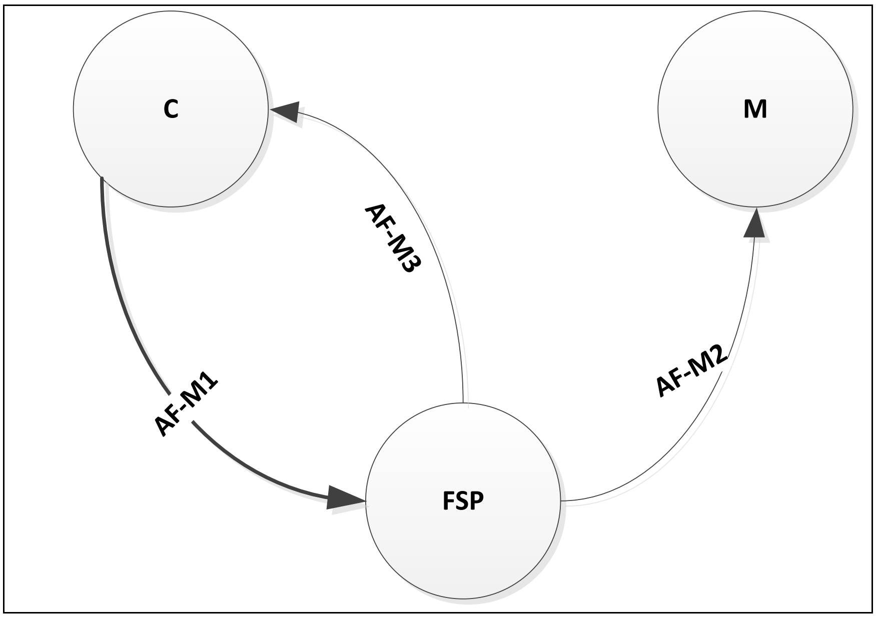


Figure 18: Dispute resolution

4.3. Dispute Analysis

At the end of exchanging a digital commodity and a payment (between the two transacting parties, the Merchant (M) and the Customer (C)), there are three possibilities for C (here we consider a normal exchange where no protocols are used):

1) **C** obtained the correct digital commodity

2) **C** obtained an incorrect digital commodity

3) **C** did not obtain any digital commodity

There are also three possibilities for **M**:

1) **M** received the correct payment

2) **M** received an incorrect payment

3) **M** did not receive the payment at all

The incorrect digital commodity scenario means that the obtained digital commodity is not the one that **C** had ordered, whereas the incorrect payment scenario means that the received payment is not the same as the price requested by **M**.

Table 4 illustrates the combination of these possibilities for **C** and **M**. In Table 4, X denotes that each Participant (C or M) has not obtained the requested product under no conditions, or each Participant has obtained an wrong product; while √ denotes that the right product has been obtained. observe that the decision of the resolve is not particular, as this discussion here is concerned with exchange and resolution in general[4].

	C	**M**	**Result**
	Receives digital product	Receives payment	
1	√	√	No dispute
2	√	X	**M** disputes
3	X	√	**C** disputes
4	X	X	There are possibilities for disputes by both **C** and **M**

Table 4: Dispute possibilities

	C	**M**	**Result**
	Receives decryption key	Receives payment	
1	√	√	No dispute
2	√	X	Not applicable
3	X	√	**C** disputes
4	X	X	No dispute / Not applicable / **C**’s fault

Table 5: Disputes possibilities for the proposed protocol

In the above tables, we realise that if **M** and **C** obtain each other’s items, then there is no need for a dispute (case 1 in Table 4). If **C** obtains the ordered digital commodity and **M** either obtains an incorrect payment or does not receive the payment at all, then **M** will dispute (case 2 in Table 4). On the other hand, if **M** obtains the correct payment and **C** either obtains a wrong product or does not receive the product at all, then **C** will dispute (case 3 in Table 4). Finally, case number 4 in Table 3 has four possibilities, which are as follows:

1. If both entities (**C** and **M**) do not obtain each other's items, then there will be no dispute, as no one is disadvantaged.
2. If both entities (C and M) gain wrong products every party (i.e. C obtains a wrong digital commodity and **M** obtains an incorrect payment), then each party **C** and **M** will dispute.
3. If **C** obtains a wrong product and **M** does not obtain the payment absolutely, then both parties (**C** and **M**) will dispute.
4. If **M** obtains an incorrect payment and **C** does not obtain the digital commodity at all, then both entities (**C** and **M**) will dispute.

For the proposed protocol (Table 5), the two parties (**C** and **M**) actually exchange a payment (from **C**) and the decryption key for the encoded digital commodity (from **M**). This is because **M** transmits the encrypted digital commodity to **C**, and **C** verifies it; if satisfied, the two parties then formalize the exchange of the payment and the decryption key. The correct order of the exchange process of the payment and the decryption key in the proposed protocol is that **M** obtains a correct payment before **C** obtains the decryption key.

The following scenarios study the cases presented in Table 5, which presents all the possible scenarios for dispute in the proposed protocol (the meanings of X and √ in Table 5 are the same as the ones in Table 4).

1. In scenario 1, both **C** and **M** obtain the correct product that they have ordered (i.e. **C** obtains decryption key and **M** obtains the correct payment). Therefore, there is no dispute.
2. In scenario 2, **C** obtains a right coded key, and **merchant** obtains a wrong payment or does not obtain the payment absolutely. The scenario is not relevant in the proposed protocol, as **customers** have to submit a right payment in order to obtain the right coded key.
3. In scenario three, **customer** either obtains a wrong coded key or does not obtain the decryption key absolutely, and **merchant** obtains the right payment. In such situation **customer** should lodge a complaint to the **FSP**.
4. In scenario 4, the potentials are:
    a) Neither party (**C** nor **M**) obtain each other's items. Therefore, there will be no dispute, as they have not disclosed their items to each other.

    b) Parties (**C** and **M**) obtain the wrong product from each other, i.e. **C** receives the wrong coded key and **a merchant** obtains a wrong payment. The scenario is not relevant in this proposed protocol; as **customers** have to transmit the right payment for obtain a correct decoded key. Hence, if **M** discovers that the wrong payment has been sent, after that merchant shouldn't submit a decoded key absolutely.

    c) **C** obtains a wrong decoded key and **merchant** doesn't obtain a payment absolutely. The scenario is not relevant in this proposed protocol, as **customers** have to transmit the right payment in order to obtain the correct decryption key. As well, if **merchants** haven't obtained a payment, thereafter **merchant** shouldn't deliver a decoded key absolutely.

d) **Custmoer** doesn't obtain a decoded key and **merchant** obtains the wrong payment. Here is a common situation, as **merchant** shouldn't submit the decoded key to **customer** if a payment is incorrect. In other words, **C** has to deliver the right payment for obtain a correct decoded key from **M**. Therefore, in this situation, **C** raises a dispute with the **FSP**. **C** must deliver the correct payment to the **FSP**. Whether **customer** delivers the right payment to the **FSP**, thereafter the **FSP** should organize solve for each parties, **customer** and **merchant**. **M** can undoubtedly decide to ignore the incorrect payment but if the **FSP** discovers that the wrong payment has been sent, then **C**'s request for a dispute will be refused.

The proposed protocol has been formulated in a way that minimizes disputes. In addition, **C** is the only party that will ask for solve, as **merchant** should not deliver an item until the correct product is received from **C**.

For extention to the above scenarios, the next situations are also discussed.

- **Customer** ask for solve that after decrypting the digital product, he finds the item is wrong. This scenario is impossible, as **FSP: PK** ensures that the item is right; also, **Customer** should not have delivered the payment to **M** if he discovers that the wrong **D** has been sent. Therefore, it is **Customer**'s responsibility (to delivering a payment to **merchants** when he had reservations for the item). At the moment when **customer** delivers a payment to **merchant**, it is a clear indication that **C** is satisfied with **D,** and hence **C** cannot raise any dispute because the protocol rules allows customer to verify the commodity prior sending the money to **merchant**.

- Although it is quite evident **merchant** shouldn’t ask for solve , as **merchant** obtains the payment before delivering the decoded key to **customer**, the next situation are thoughtful:
  - **M** alleges that he has obtained the wrong payment from **C**. This is an unlikely scenario, as **C** is aware of the protocol’s rule that if he delivers the wrong payment, he shouldn’t obtain a decoded key. But, if this case occurs, **customer**’s responsibility (for delivering the incorrect payment). Hence **M** will not deliver the decryption key if he obtained the incorrect payment, i.e. **M**’s item is not revealed and hence **M** does not need to dispute[4].

4.4. Analysis of Scenarios

When implementing the proposed protocol between **C** and **M**, there are various possible scenarios, which include the following:

a) Parties, **consumer** and **merchant**, are acting fairly.

b) **Consumer** is acting unfairly, and **merchant** is acting fairly.

c) **Merchant** is dishonest and acting unfairly, while **consumer** is acting fairly.

d) Each parties **Consumer** and **Merchant** are dishonest and acting unfairly.

The possible scenarios for executing the protocol are as follows:

1) Both parties, **Consumer** and **Merchant**, are fair in their dealings, which leads to normal implementation of the protocol, where **M** submits a correct M1, **C** submits a correct M2, and **M** submits a correct M3 (Figure 19).

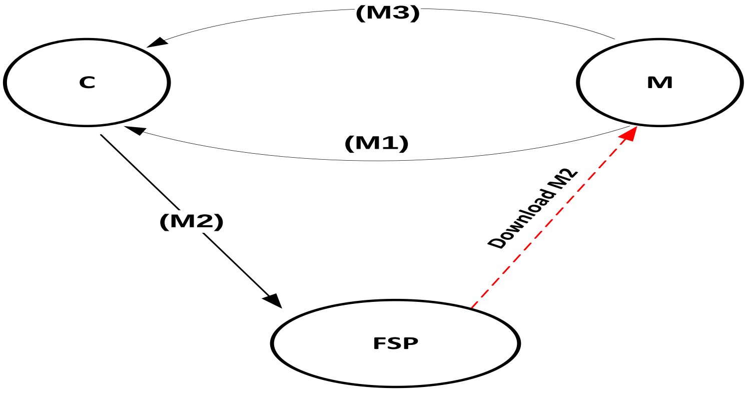


Figure 19: Proposed Protocol, Scenario 1

2) **C** aborts the protocol after obtaining M1 because M1 is either wrong or **C** has lost interest in the exchange process (Figure 20).

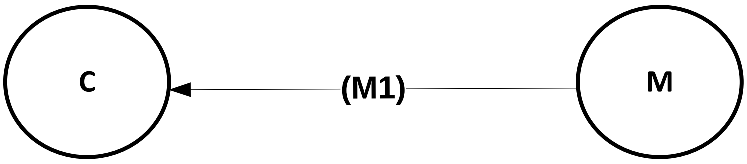


Figure 20: Scenario 2

3) After obtaining M1, **C** communicates with the **FSP** before sending M2. If, the **FSP** discovers that AF-M1 is wrong, then the **FSP** transmit a revoke message to

C. In this case, **C** tried to behave dishonestly however did not achieve anything (Figure 21).

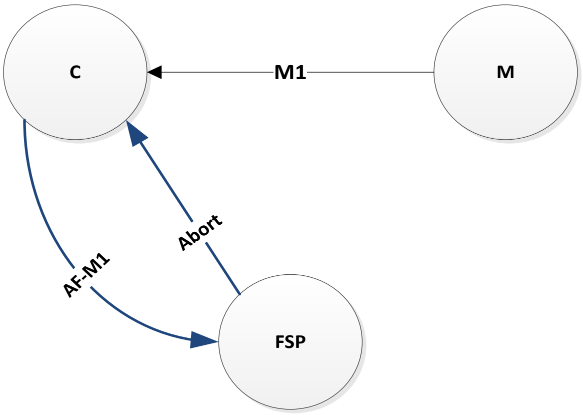


Figure 21: Scenario 3

4) The same as scenario number 3 but in this situation, the **FSP** found that AF-M1 is correct. Hence, the **FSP** will try to create a fair exchange for **C** by sending a warning to **M** in AF-M2 in addition to submitting the saved tSK to **C** in AF-M3; **M** will not gain any advantage over **C** (Figure 22).

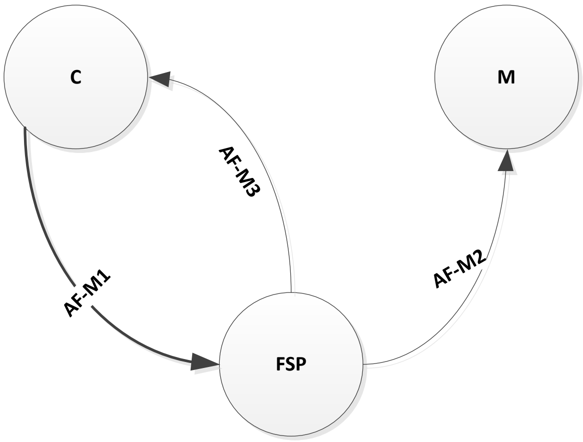


Figure 22: Scenario 4

5) **C** obtains M1 from **M** who discovers that M1 is right, and after that **C** transmits M2 to the **FSP**. **Consumer** waits for M3 from **M** but obtains nothing. Hence, **C** communicates with the **FSP** for settlement. But, if the **FSP** discovers that AF-M1 is wrong, it transmits a revoke message to **C**. There are two situations as to wherefore **merchant** failed to send M3 to **C**. These possibilities are either because **M** discovered that M2 was wrong or because **M** is dishonest in his dealings. If M2 is incorrect, then it is **C**'s responsibility (for sending an incorrect M2 to **M**). In the latter scenario (where M2 is correct but **M** is dishonest), then **C** needs to send AF-M1 to the **FSP** in order to obtain a resolution (Figure 23).

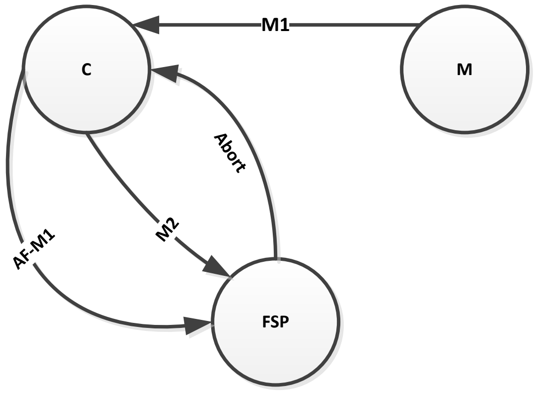


Figure 23: Scenario 5

6) After **C** receives M1 from **M**, **C** discovers that M1 is right, and then **C** transmits M2 to the **FSP**. **M** then discovers that M2 is right and transmits M3 to C. However, C communicates with the **FSP** for resolution after obtaining M3. If the **FSP** discovers that AF-M1 is wrong, it transmits a revoke message to **consumer**. In this scenario, **consumer** communicated with the **FSP** for two possible reasons; the first is that M3 is wrong, and another is that M3 is right

However **C** intends to find out which he able to obtain from the **FSP** in order to gain an unfair advantage over **M**. However, in both possibilities, **C** must deliver a correct AF-M1 in order to achieve a resolution (Figure 24).

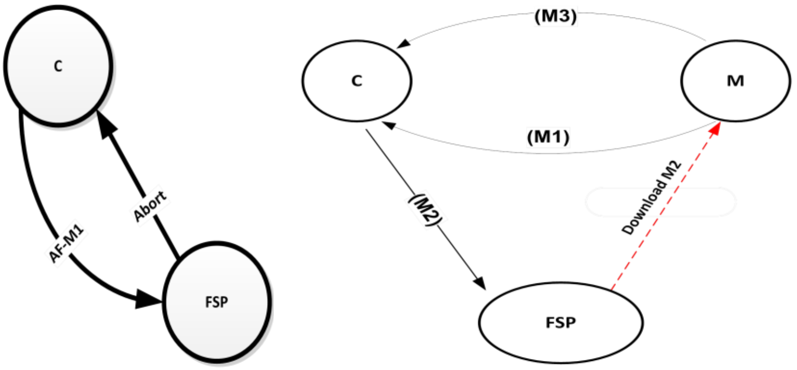


Figure 24: Scenario 6

7) The same as scenario number 6 However there, the AF-M1 which **C** transmits to the **FSP** is right. Hence, the **FSP** solve it by submitting AF-M2 to **M** and AF-M3 to **C** (Figure 25).

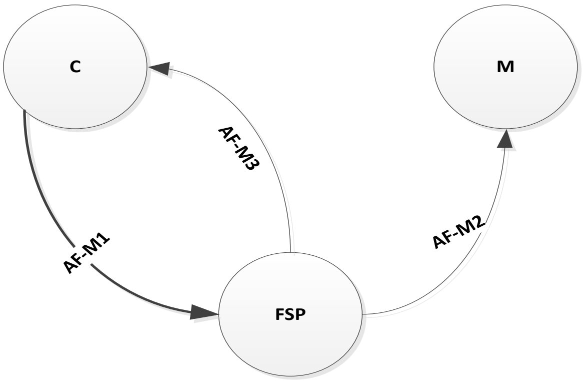


Figure 25: Scenario 7

Therefore, it is clear from the above scenarios, **Consumer** and **Merchant** are achieve the fairness. On the other hand, the following scenarios study where the messages (M1, M2, M3, AF-M1, AF-M2 and AF-M3) of the proposed protocol are sent but have not been received because of a failure in the communication channels between the parties involved in the protocol:

1) If M1 is not received by **C**, then fairness will not be compromised because no one has revealed their item; **C** will not send M2 if he has not received a correct M1.
2) If M2 is not received by the **FSP**, then **M** will consider that **C** is not interested in the exchange, and hence **M** will not send M3. As a result, **C** will wait for M3 from **M** and as **M** has not obtained M2 from **C**, **M** will not submit M3. Therefore, **C** will contact the **FSP** for resolution. However, there is the possibility that the communication channel failed; this possibility will be studied later.
3) If M3 is not received by **C**, then **C** will consider it as **M** behaving dishonestly. As a result, **C** will contact the **FSP** for resolution. However, there is the possibility that the communication channel failed.
4) If AF-M1 is not received by the **FSP**, then **C** needs to re-contact the **FSP**, again asking for a resolution.
5) If AF-M2 is not received by **M**, then there are two possibilities:
    a. If AF-M2 is not received by **M** (and **M** has not received a correct M2) and at the same time AF-M3 is not received by **C**, then fairness is not compromised.

b. If AF-M2 is not received by **M** but AF-M3 is received by **C**, then fairness is compromised if, and only if, **M** has not received a correct M2.

6) If AF-M3 is not received by **C** then there are two possibilities:

a. If AF-M3 is not received by **C** and at the same time AF-M2 is not received by **M** (and also **M** has not received a correct M2), then fairness is not compromised.

b. If AF-M3 is not received by **C** but AF-M2 is received by **M**, then fairness is compromised if, and only if, **C** has not received a correct M3.

As can be seen in these cases, the channels of communication among all parties should be resilient for fairness to be ensured. To ensure fairness even in the case of communication channel failure, fault tolerance techniques need to be applied to the proposed protocol. However,we will leave this techniques for exrended this work.

4.5. Comparisons of Protocols

Only three messages are required for exchange the item, and these three messages are also used in the case of disputes.

The way in which products (digital or physical) and payments are exchanged online is that a customer delivers the money to a seller, after the seller confirms the correctness of the money. If the payment is right, the seller delivers the item to the buyer. This method is applied in most protocols (MP).

However, the method that is applied in our protocol is that the merchant delivers an encrypted digital commodity to the customer, and then the customer verfiy the validity of the coded item. If the buyer is happy with the digital item, then he delivers the money to the merchant.

Each has advantages and disadvantages. The advantage of the method applied in most protocols is that it follows the conventional order of exchange, where a buyer transmits a money first, after that the seller sends a product. Therefore, neither customers nor merchants will feel that there has been any change in the way they conduct business. On the other hand, they will notice a difference in our protocol, as the merchant begin the process by transfering the items.

Additionally, in the MP, the merchant receives the first message (M1), which contains the encrypted payment, but then the merchant may decide not to complete the exchange, i.e. not to send the second message (M2) to the customer. However, in our protocol the customer receives the first message (M1), which contains the encrypted digital product, but then the customer may decide not to complete the exchange, i.e. not to send the second message (M2) to the merchant.

Therefore, if our protocol and the MP are compared in this respect (i.e. the party who receives M1 decides not to complete the exchange because either they are not interested in the exchange or M1 is incorrect), then our protocol would be better in that it involves less encryption.

The MP places more responsibility on the merchant to find a digital product that matches customer’s requirements. In this way, the merchant is forced to be fairly, and transmits the right digital commodity to the customer so that he can obtain the decryption key in order to decrypt the encrypted payment[4].

Our protocol, on the other hand, places more responsibility on the customer to deliver the right price for obtain the decoded key to decode the coded digital product. In this way, the customer is forced to be honest.

In the case of disputes, the total size of the messages in the dispute resolution phase for MOP is greater than the size of the messages (i.e. AFM1, AF-M2, and AF-M3) in the dispute resolution phase for our protocol. The reason for this is that in MP, the encrypted digital commodity is included in at least two messages, while in our protocol; the encrypted digital is included in message AF-M2 only.

The proposed protocol	# messages in process phase	# messages in solve problems phase	Starts the exchange
	3	**3**	**M**
	Party who raises disputes	# of modular exponentiations (process phase)	# of modular exponentiations (solve problems phase)
	C	**12**	**6**

Table 6: Features of our protocol

4.6. The Timing of the Protocol

The timing of the execution of the new protocol is debated in this part (the timing numbers are in nanoseconds).

The specifications of the computer used to perform and measure these timings are as follows. The processor is Intel(R) Core(TM) i5 CPU M460@ 2053GHz 2053 and 4GB of RAM. For the protocol, three scenarios are carried out. The first one has a digital product (D) sized 28KB, the second one has a digital product (D) sized 2.2MB, and the third one has a digital product (D) sized 10MB.

ID	Action	Scenario 1: digital product size = 28 KB	Scenario 2: digital product size = 2.2 MB	Scenario 3: digital product size = 10 MB
Pr-M1	*Generating temporary session key*	14	35	105
Pr-M1	*Encrypting temporary session key*	12	38	114
Pr-M1	*Sending encrypted temporary session key*	23.7	239	717
Pr-M1	*Sending invoice details*	4	4	12
Pr-M2	*Generating message digest for signing invoice*	3	29	87
M1	*Encrypting product using session key*	34	53.6	160.8
M1	*Encrypting the product, invoice and FSP PK*	20	98	294
M1	*Sending the product, invoice and FSP PK to the customer*	147	537	1611
M1	*Decrypting merchant message*	47	59	177
M2	*Validating invoice received from customer*	3	3	9
M3	*Sending session key to customer*	61	78	234
Total		***368.7***	***1173.6***	***3520.8***

Table 7: Timings of the proposed protocol

In the table above, we can see that the time is longer when a digital product is tranmit with the message. The time needed to construct message M1 of the proposed protocol (see table 7), when the digital product size is 2.2MB, is greater than the time needed when the digital product size is 28KB.

4.6.1. Cost functions for the proposed protocol

In this section, a cost function is devised for the proposed protocol. The cost function to be devised concerns with the time needed to form/construct the messages of the exchange phase of the protocol. There are three messages required, and therefore, a cost function is devised for each message; then, a general cost function is devised for the all the messages of the protocol.

M1:

The message M1 includes:

- The product is coded using the tsk.
- The whole message is coded using the SK of the merchant.
- The message consists of the Financial Service Provider's public key.
- That public key has the information about the service provider to whom the consumer should make the payment.
- Invoice: [Product information, Merchant ID].

M1: Merchant → Consumer

M: SK S [tSK E[D], N, FSP:PK, FSP: SK S [N]].

N: [Di, Merchant ID, C].

M will need to construct the items (***SK S, tSK E[D], N, FSP: PK, FSP: SK S [N]***). So, the cost function for M1 is the sum of the time needed to construct these items:

f(M1) = SKS + tSK E[D] + N + FSP: PK + FSP: SK S [N].

The time needed for constructing ***tSK E[D]*** will change according to the size of the digital product (D), whereas the time needed to construct the other items will remain roughly the same because the size of the keys will remain the same in any message, even if the keys change. Therefore, the cost function for M1 is:

f(M1) = c1 + tSK E[D] + c2 + c3 + c4

Where c1, c2, c3 and c4 are constants that represent the time needed to construct ***SKS, N, FSP: PK, FSP: SK S [N]***, respectively.

M2:

The message M2 includes two items, namely, ***Payment Information (Pi)*** and ***N***. Therefore, the cost function for M2 is the sum of the time needed to construct these two items:

f(M2) = Pi + N

The payment normally includes specific information (such as the names of the payer and payee, and the amount) which has roughly the same size even if the information included in the payment is different. Therefore, the time needed to construct the two items included in M2 will remain roughly the same in any message. Therefore, the cost function for M2 is:

f(M2) = c5 + c6

where c5 and c6 are constants that represent the time needed to construct ***Pi*** and ***N***, respectively.

M3:

The message M3 includes one item, which is ***SK S [tSK].*** Therefore, the cost function for M3 is:

f(M3) = SK S [tSK].

The time needed to construct M3 will remain roughly the same even if the session key changes because the size of the keys will be the same. Therefore, the cost function for M3 is:

f(M3) = c7

where c7 is a constant that represents the time needed to construct ***SK S [tSK].***

4.6.2. The cost function for the proposed protocol:

The cost function for the proposed protocol is the sum of the cost functions for M1, M2 and M3. That is:

f(proposed protocol) = f(M1) + f(M2) + f(M3)

f(proposed protocol) = (c1 + tSK E[D] + c2 + c3 + c4)+ (c5 + c6)+ (c7).

f(proposed protocol) = tSK E[D] + c1 + c2 + c3 + c4 + c5 + c6 + c7.

f(proposed protocol) = tSK E[D] + C （where C is the sum of all constants)

The time needed to construct the messages will increase significantly as the size of the digital product increases. Otherwise, the time will remain roughly the same if the size of the digital product remains the same. The time needed to construct the messages of the exchange phase of the proposed protocol under three scenarios was given in Table 7.

4.7. Comparisons with Other Protocols

The comparisons will be made in accordance with a number of different properties; (1) How many massages required in every phase?, (2) Is the trusted third party save the item?, (3) Which party will arise the dispute?, and (4)compare the modular exponentiations numbers .These different criteria will help improve the protocol by reduce the time for send and receive the messages, Reduce the time for generates and verify the messages and the use of TTP is only for verifying the items not use for delivering or store the items that will be exchange will be reduce the load on network.

To make fair comparisons, three of the protocols to be compared with ours have the same characteristics; these are Zhang *et al*. [30], Devane *et al*. [32], Q. Zhang *et al*.

. Table 8 below represents a summary

Protocol	Items to be exchanged	Fairness	Weak-nesses	Load on TTP	Efficiency	# messages (exchange phase)	Type of fairness
Zhang *et al.*	Payment and a product (digital or physical)	Yes	1	High	Medium	7 + physical delivery and collection	Strong
Devane *et al.*	Payment and digital product	Yes	1	High	Medium	7	Strong
Zhou and Gollmann	Provide the originator and the intended recipient with evidence after an execution	Yes	3	High	Medium	5	Fairness is not ensured
Q. Zhang *et al.*	Payment and digital product	Yes	1	High	Medium	12	Strong
Our Protocol	**Payment and digital product**	**Yes**	**0**	**Low**	**High**	**3**	**Strong**

Table 8: Comparison of fair exchange protocols

The proposed protocol has the least number of messages (among the other protocols) required to be exchanged. With regard to dispute resolution, all the protocols apart from the Ray *et al*. protocol [33] have the same numbers of messages required in the dispute resolution phase. To measure the performance of the proposed protocol, we performed a simulation of the Zhang *et al*. protocol [43] and the Devane *et al*. protocol [44], and compared the results of each one with our protocol. The purpose of performance evaluations is to measure the computation time of each message, and to determine the total computation time. The comparison results show that the new protocol is widely efficacious than other protocols By testing our protocol, our work has proved that the process of the proposed protocol is workable. When it is implemented in the real world, the length of the asymmetric keys becomes the main factor affecting the protocol's performance; in other words, increasing the length of the asymmetric keys (to generate the highest possible level of security) will increase the computation time significantly.The Ray *et al*. protocol allows the TTP to keep the merchant's product before the exchange process between the customer and the merchant occurs. This creates a burden on the part of the TTP, as the TTP is then responsible for storage and security assurance. According to the Ray, the TTP must contact both parties (customer and merchant) if either party has a dispute; however, in the other protocols, the polemicist and the trusted third party are engaged only.

Engaging all Participants in dispute resolution could need a lot of messages to exchange items, so the creating communication overload may be occure. Table 9 below compares all the various protocols and it clearly shows how the proposed protocol has the less number of messages and less number of modular exponentiations. Thus, the new protocol is a widely effective and efficacious protocol. When we say "more efficient" we mean that:

- Reduce the communication overload and solve bottleneck problems by:
- Reduce the numbers of messages to be exchange between the parties.
- Reduce the time for constructing the messages.
- Reduce the time for generates and verify the messages.
- The use of TTP is only for verifying the items not use for delivering or store the items that will be exchange.
- guarantee strong fairness

	Ray *et al.* 2000	**Devane *et al.***	**Ray *et al.* 2005**	**ZHA**	**Zhang *et al.***	**Q. Zhang *et al.***	**Our Protocol**
# messages in the process phase	6	7	4	4	7	12	**3**
# messages in solve problems phase	Not specified	Not specified	3 to 5	3	Not specified	Not specified	**3**
TTP type	Inline	Online	Offline	Offline	Online	Online	**Online**
TTP holds item	Yes	No	Yes	No	Yes	Yes	**Yes**
Each side ingaged in solve problems	Not specified	Yes	Yes	No	Yes	Yes	**No**
# of modular exponentiations in the process phase	20	28	27	20	Not specified	Not specified	**12**
# of modular exponentiations in the in solve problems phase	Not specified	Not specified	5 to 6	6	Not specified	Not specified	**6**

Table 9: compares with inline, offline and online protocols.

Chapter 5

Formal Verification of Protocol

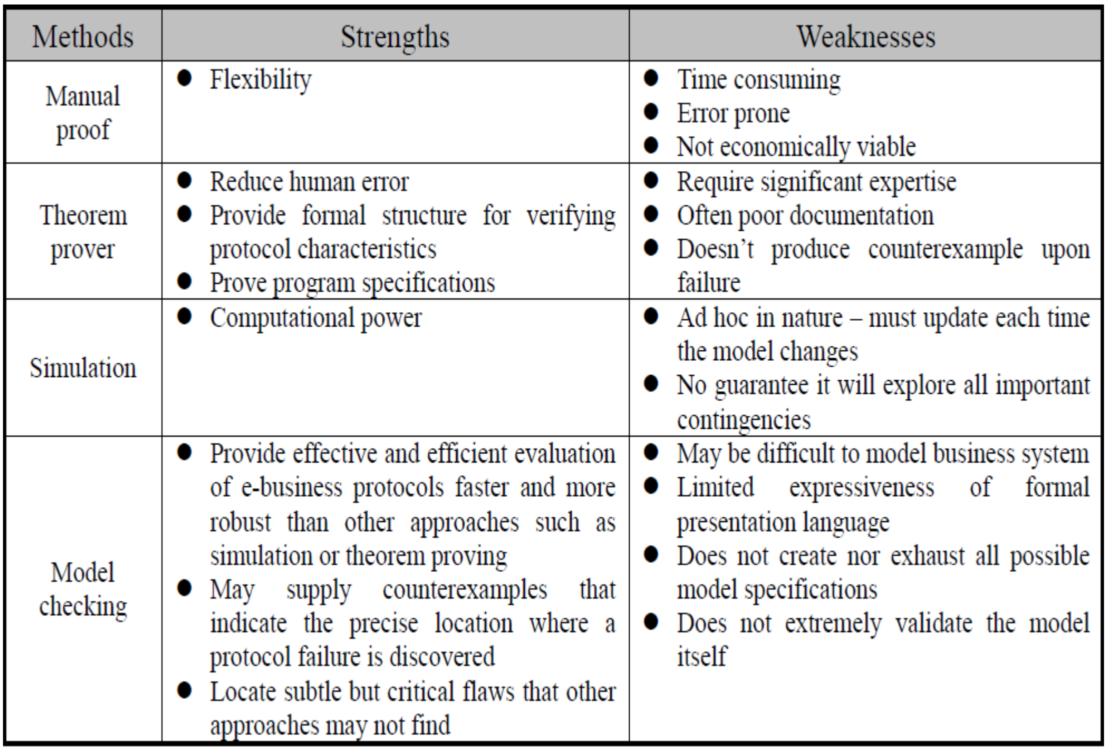
Objectives:

- *Formal Analysis (verification)*
- *Model Checking*
- *Spin Model Checker*
- *Modelling the Protocol*

This chapter discusses the formal verification process of protocols. Model checking will be used to authenticate the fairness property.

5.1. Formal Verification of Protocol

It is vital to use formal analysis during the verification process in order to discover any unforeseen errors in the design of the system. Several methods can be adopted in executing the formal process of verification, and these include manual proofs, theorem proving, and model checking [65, 66]. Some of the drawbacks associated with manual proofs are that they take a long time to execute, are slow and tend to be error-prone [65, 66]. A human user is required in theorem proving; if the theorem prover discovers a fault, then no detailed analysis of the primary cause of the glitch in the system is needed. Formal methods used for the purpose of protocol or system validation usually adopt the following technique: in the first two initial steps, high-level formal notations are used in order to (1) make a formal analysis, (2) identify officially the features with will be assessed. The formal verification process of the protocol is often long, complicated and error-prone. In the third step, the protocol is authenticated against its attributes through an automated verification tool that can read these notations. When assessing the security protocols of a system, various verification techniques can be used, and these are summarized below.



Methods	Strengths	Weaknesses
Manual proof	● Flexibility	● Time consuming ● Error prone ● Not economically viable
Theorem prover	● Reduce human error ● Provide formal structure for verifying protocol characteristics ● Prove program specifications	● Require significant expertise ● Often poor documentation ● Doesn't produce counterexample upon failure
Simulation	● Computational power	● Ad hoc in nature – must update each time the model changes ● No guarantee it will explore all important contingencies
Model checking	● Provide effective and efficient evaluation of e-business protocols faster and more robust than other approaches such as simulation or theorem proving ● May supply counterexamples that indicate the precise location where a protocol failure is discovered ● Locate subtle but critical flaws that other approaches may not find	● May be difficult to model business system ● Limited expressiveness of formal presentation language ● Does not create nor exhaust all possible model specifications ● Does not extremely validate the model itself

Table 10: Summary of formal verification methods. [67].

In this research, we will use model checking for the formal verification of the fairness property of the protocols, using a model checker called SPIN.

5.2. Model checking

Model checking is a formal method for verifying finite-state concurrent systems. The description of the system to be authenticated is expressed as a suitable dialectics (normally a temporal dialectics like as computational tree logic(CTL) [68], linear temporal logic(LTL) [69], and alternalting time temporal logic (ATL) [70]). It confirms if the fromwork fulfils this characteristic by looking into each the likely activities of the framwork. Raskin et al have tried to prove in [71] wherefore model checking is the best formal method for the authentication of fair exchange protocols. This method is the best suitable because the model protocols are treated as games and their participants as players. Fair exchange protocols are composed of many sub-protocols that can be applied at various times when carrying out a protocol. Hence these protocols are not linear, contrary to key exchange and authentication protocols.

The sub-protocols are supposed to be carried out in a predefined sequence. If they are carried out contrary to the protocol designer's predefined order, a possible security breach may occur. The key feature of layout a protocols as play will be any attempt to disrupt the operation of the system can be taken into consideration. During every stage of the play (the process of the protococl), every entity or player (protocol parties) can make more than one move. A parties must follow the protocol by sending the precise message as specified by the protocol, but he or she can pose a threat by transmitting a message that is not consistent with the original one. These messages that are inconsistent with the original protocol can be designed by merging parts of previous messages and then applying different computations.

The model checking method is used to confirm that the protocol fulfils a specific property. It examines all feasible operations of the game (a model of the protocol). This means that it confirms the feasible combination of all the moves of the participants (players). Regardless of the moves, the property will always hold as far as it has been authenticated by a model. We can verify the fact that the fairness of the protocol cannot be violated by any dishonest player as far as the protocol fulfils the fairness property.

5.2.1. SPIN Model Checker

This model checker was created in the 1980s by Holzmann at Bell Laboratories. It is one the most popular tools and was given in 2001 the ACM's prestigious System Software Award [72]. SPIN is a tool that is free and open source. SPIN is work here to authenticate the fairness feature of the protocol mentioned in this thesis. The main reason for choosing SPIN is that despite being a model checker it is also a simulation tool. Thus, it aids in simulating the assumptions of the suggested protocol. The second reason for adopting SPIN is that it can successfully validate fair exchange protocols [73, 74]. The third reason is that the modelling language that is recognised by SPIN for model specification resembles the C language, which is a language that is simple to learn. Fourthly, SPIN is a powerful free and open source tool, which is widely available. When applying SPIN to authenticate the fairness property of a system, the authentication language, Promela (Process Meta Language) [72], must be used to identify the model. In addition, the fairness property of the protocol to be authenticated is identified using LTL (Linear Temporal Logic) [72]. After the model and the properties have been successfully verified, they are fed into SPIN's GUI (Graphical User Interface), which is known as iSPIN.

If there is any syntax error, iSpin will detect it. If no errors are found at all in the specification, then SPIN will confirm whether the specified property has been authenticated against the specified model. If the result confirms that the verification is valid, it will be indicated on the GUI. On the other hand, if the verification of the property fails, a counterexample will be given, which will detect the cause of the error in the model.

• *Promela*

Promela (Process Meta Language) is the language of authentication that is understood by SPIN. There are three main objects that any Promela mode is made from [4]:

- Processes
- Data objects
- Message channels [72]

Each entity of a system is treated and modelled as a process. A process is used to define the behaviour or activities of the parties. A process is defined using the keyword *proctype*, followed by the process's name and a set of parameters. The following table is an example of the process:

```
  proctype Consumer(chan channel_C_M, channel_C_FSP)
 {
bool sendM2, send AF-M1;
        sendM2 = FALSE;
        sendAF-M1 = FALSE;
   }
```

When we have identified the process, we can then instantiate the process by using the keyword *run*. There are two kinds of variables for the basic data objects: local variables, which are defined within the scope of a process, and global variables, which are defined outside the scope of a process. The basic data types used in Promela are: *bit*, *byte*, *bool*, and *int*. The message channels are used to model the transfer of data from one process to another. They are declared using the keyword *chan*. For example:

```
chan ch = [0] of {bit};
```

Communication through message channels can be defined as a synchronous message channel, and each message contains one field which is of *bit* type. The statement below sends a message with the value of C, which is 0, to the channel *ch*;

```
bit c = 0
ch ! c ;
```

The statement below obtained a message from channel *ch* and save the amount in changeable m:

```
ch ? m;
```

Promela statements are either executable or blocked. A statement is executable as far as it is not blocked, but once it is blocked, it will not be executable. Print statements and assignment statements are always unconditionally executable. An expression statement is executable only if it is non-zero (true). If the expression value is zero, then the condition is false.

- ***LTL***

LTL (Linear Temporal Logic) [72] is used by SPIN for describing behavioural the properties of a system. In other words, it checks the model's properties to be authenticated. The operators of LTL describe a pattern of the order of events on a single computation path [75]; that is, the operators describe a pattern for a sequence of events in a set of states of a system.

The following diagram explains the above statement[4]:

"and" (∧), "or" (∨), "not" (¬), and "implies" (⇒);

LTL provides "always" (□), "eventually" (◊), and "next time" (○).

5.3. Modelling the Protocol

The SPIN model checker is a system that is used to check the fairness Characteristic of proposed protocol. When using the SPIN model checker for the authentication of the fairness property of the protocol, a protocol meta-language, Promela, is used to model against the specified protocol, and the fairness property specified using LTL. All the participants in the protocol are represented by a process in Promela, and their activities are also specified in the process. The various entities that take part in the protocol are Customer, Merchant and FSP (Financial Service Provider).

5.3.1. Modelling the Customer process

The Customer as a party in the fair exchange protocol is represented by the Customer process. The process begins as the Customer waits for the message M1 from the Merchant process. All feasible activities of the Customer are listed once message M1 is obtained (Chapter 5 discusses all feasible activities).

The possible activities are: M1 is accurate and the Customer wishes to continue the exchange; M1 is accurate but the Customer does not wish to continue with the exchange; or M1 is wrong but the Customer communicates with the FSP for resolution before transmitting M2 to the FSP. Each and every likely activity of the Customer is indicated in the Customer process. Hence, these activities together with all probable activities of the Merchant and the FSP are authenticated by the model checker. Similarly, all probable activities of the Customer are identified after transmitting M2, obtaining M3, transmitting AF-M1, and obtaining AF-M3.

The diagram below partly illustrates all likely activities of the Customer after receiving message E-M1.

```
 /* Consumer

Sends messages: M2, send AF-M1

Receives messages: M1, M3, AF-M3


*/

proctype Consumer(chan channel_C_M, channel_C_FSP)

{


        bool sendM2, send AF-M1,waitForM1, waitForM3, waitForAF-M3;


        sendM2 = FALSE;

        sendAF-M1 = FALSE;

        waitForM1 = FALSE;

        waitForM3 = FALSE;

        waitForAF-M3 = FALSE;

        quitConsumer = TRUE;

        do

        :: (waitForM1 == FALSE) ->

                channel_C_M!M2;

                sendM2 = TRUE;

                waitForM1 = TRUE;

                quitConsumer = FALSE;

                printf("Consumer: M2 sent to FSP \n");
```

```
:: (quitMerchant == FALSE && waitForM1 == TRUE &&
                waitForM3 == FALSE) ->
                 channel_C_M!M1;
                 sendM2 = TRUE;
                 waitForM3 = TRUE;
                 printf("Consumer: M2 sent to Finacial Service Provider \n");
                 if
                         :: TRUE -> /* Consumer like to go-ahead   */
                         sendM2 = TRUE;
                         quitConsumer = FALSE;
                         printf("Consumer: Consumer like to go-ahead \n");

                         :: TRUE -> /* Consumer do not want to go-ahead   */
                         quitConsumer = TRUE;
                         printf("Consumer: Consumer do not want to go-ahead  \n");
                         break;
```

5.3.2. *Modelling the Merchant process*

The Merchant as a party in the fair exchange protocol is represented by the Merchant process. The process begins when the Merchant transmits the message M1 to the Customer process. After the message is transmitted, the process then waits for the message M2 to be obtained from the FSP process. Upon obtaining M2, the Merchant process will specify all likely situations and activities of the Merchant.

The diagram below partly illustrates the Merchant process, which indicates transmitting message M1 to the Customer process.

```
          /* Merchant
         Sends messages: M1, M3
       Receives messages: M2, AF-M3
     */
proctype Merchant(chan channel_C_M, channel_M_FSP)
{

        bool sendM1, sendM3, waitForM2, waitForAF-M3;

        sendM1 = TRUE;
        sendM3 = FALSE;
        waitForM2 = FALSE;
        waitForAF-M3 = FALSE;
        quitMerchant = FALSE;
        do
```

```
            :: (sendM1 == TRUE) ->
               channel_C_M!M1;
                sendM1 = TRUE;
               quitMerchant = FALSE;
               printf("Merchant: M1 sent to Consumer \n");
                  if
                         :: channel_C_M?M2 ->
                          waitForM2 = TRUE;
                          printf("Merchant: Recieves the M2 from the FSP  \n");
                            if
                    :: TRUE -> /* Merchant decided the go-ahead with the purchase  */
                                  quitMerchant = FALSE;
                                  channel_C_M!M3;
                                  sendM3 = TRUE;
      printf("Merchant: Merchant decided the go-ahead with the purchase  \n");
               :: TRUE -> /* Merchant decided not to go-ahead with the purchase  */
                                  quitMerchant = TRUE;
               printf("Merchant decided not to go-ahead with the purchase  \n");
                                  break;
                                  :: timeout ->
                                  quitMerchant = TRUE;
                                  printf("Timeout at Merchant \n");
                                  break;
```

5.3.3. Modelling the FSP process

The FSP (Financial Service Provider) process in Promela represents the FSP entity in the proposed fair exchange protocol. Only the customer can contact the FSP for complaints. Hence, the FSP model listens for incoming messages from the Customer process. Upon obtaining the message M2 from the Customer process, the FSP process transmits the messages to the Merchant process.

```
  /* Financial Service Provider (FSP)
        Sends messages: AF-M3
       Receives messages: M2, AF-M1
     */
proctype FSP(chan channel_M_FSP, channel_C_FSP)
{
        do
        :: channel_C_FSP?M2 ->
                channel_M_FSP!M2;
printf("FSP validates the payment details and send the confirmation to Merchant  \n");
                if
                :: TRUE -> /* Payment is successfuly validated   */
                paymentValidatedSuccessful = TRUE;
                printf( "FSP: Payment is successfuly validated");
                :: TRUE -> /* Payment couldn't be validated   */
                paymentValidatedSuccessful = FALSE;
                printf("FSP: Payment couldn't be validated");
```

5.3.4. Modelling the Fairness property

Fairness in the protocol can be achieved when both transacting parties acquire each other’s items, or none do so. For example, if the exchange involves a cell phone, at abortion the exchange, the consumer will acquire a cell phone item and the seller will acquire a correct money, or neither of them will obtain anything. This fairness property is specified in the LTL formula as follows[4]:

$$\square((quit\_Merchant \wedge quit\_Customer) \Rightarrow$$

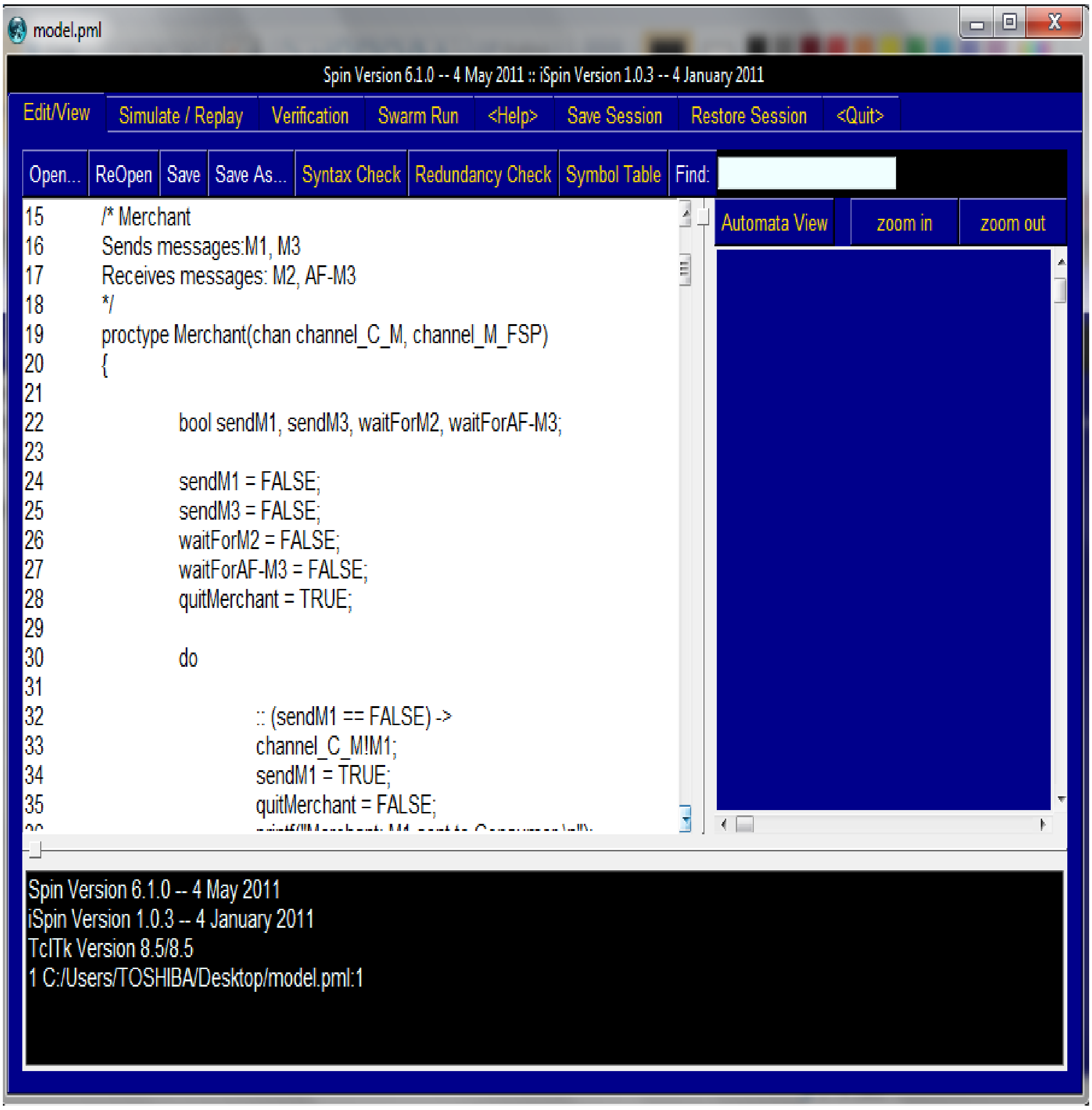
$$\lozenge((receive\_CorrectPayment \wedge receive\_CorrectProduct) \vee$$

$$(\neg receive\_CorrectPayment \wedge \neg receive\_CorrectProduct)))$$

This LTL formula shows that at the end of the protocol the two parties involved in the exchange process (the Customer and the Merchant) have exchanged each other’s items. That is, either the Customer has gain the digital product and the seller money, or neither of the parties has received anything.

5.4. The Verification results

iSPIN is used to write the protocol's specifications in Promela. Figure 26 below illustrates the iSPIN and the protocol specification in Promela.

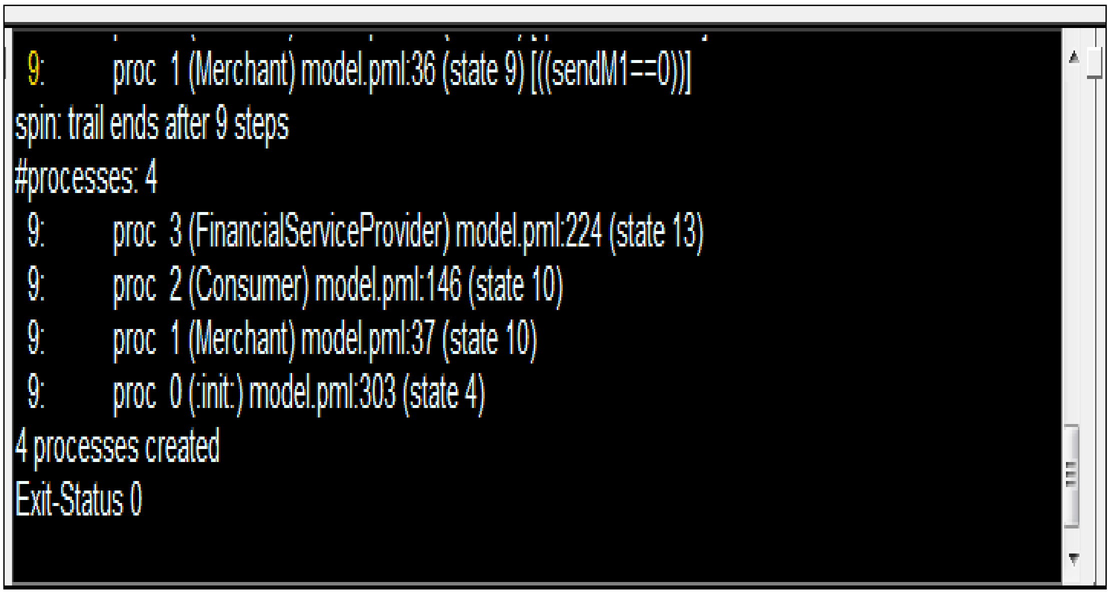


```
/* Merchant
Sends messages:M1, M3
Receives messages: M2, AF-M3
*/
proctype Merchant(chan channel_C_M, channel_M_FSP)
{

        bool sendM1, sendM3, waitForM2, waitForAF-M3;

        sendM1 = FALSE;
        sendM3 = FALSE;
        waitForM2 = FALSE;
        waitForAF-M3 = FALSE;
        quitMerchant = TRUE;

        do

                :: (sendM1 == FALSE) ->
                channel_C_M!M1;
                sendM1 = TRUE;
                quitMerchant = FALSE;
                printf("Merchant: M1 sent to Consumer \n");
```

```
Spin Version 6.1.0 -- 4 May 2011
iSpin Version 1.0.3 -- 4 January 2011
TclTk Version 8.5/8.5
1 C:/Users/TOSHIBA/Desktop/model.pml:1
```

Figure 26: protocol verification

iSPIN gives the means for simulating the protocol specified in Promela. Several simulations of the protocol are carried out in order to examine the activities of the entities involved in the transaction.



```
 9:     proc  1 (Merchant) model.pml:36 (state 9) [((sendM1==0))]
spin: trail ends after 9 steps
#processes: 4
 9:     proc  3 (FinancialServiceProvider) model.pml:224 (state 13)
 9:     proc  2 (Consumer) model.pml:146 (state 10)
 9:     proc  1 (Merchant) model.pml:37 (state 10)
 9:     proc  0 (:init:) model.pml:303 (state 4)
4 processes created
Exit-Status 0
```

Figure 27: The output shows that the processes are executed successfully.

5.5. Summary

This chapter has discussed the formal verification methods for system authentication. It has also discussed the protocol modelling and the specification of the fairness property. We have also shown that the SPIN model checker validates the specified fairness property against the specified models. The model or specification is not the same as the protocol itself. This can be caused either by errors in the modelling or by failure on the side of the specification of the protocol to fully express the protocol. The main reasons for failure in the specification include: firstly, the model of the protocol usually specifies the protocol in a general or abstract way, and hence, the model will not show all the items of the protocol but will only indicate all likely activities that the model checker will examine in order to confirm if a property holds. Secondly, there may be some weaknesses in the formal techniques used, which leads to their inability to adequately express the protocol.

The SPIN model checker therefore assists in the formal authentication of the fairness property against the specified models. It also assists in simulating all possible situations of the protocol. Another side, the execution of the protocol (to be discussed in the next chapter) is crucial and totally different from the modelling of the protocol. This is because protocol execution is mainly concerned with handling real data. The implementation of the protocol will help in dealing with real data (digital commodity and payment) and entities (C, M, FSP). It also deals with whether all entities can make and transmit the protocol's messages, validates the accuracy of the received messages, and asks for a resolution in case of any disputes.

Chapter 6

Protocol Implementation

Objectives:

- *High Level Design*
- *Activity Diagrams for the Protocol*
- *Implementation*
- *Application Control Panel*
- *Security Application*
- *User Interface Elements*
- *Performance Evaluations*

In this chapter, we discuss the design and execution of the prototype that executes the protocol.

6.1. High Level Design

The Java programming language is used to create a prototype to test the concept. The architectural design of the system for the protocol is contain of the next entities.

- FSP Server
- Merchant Server
- Customer

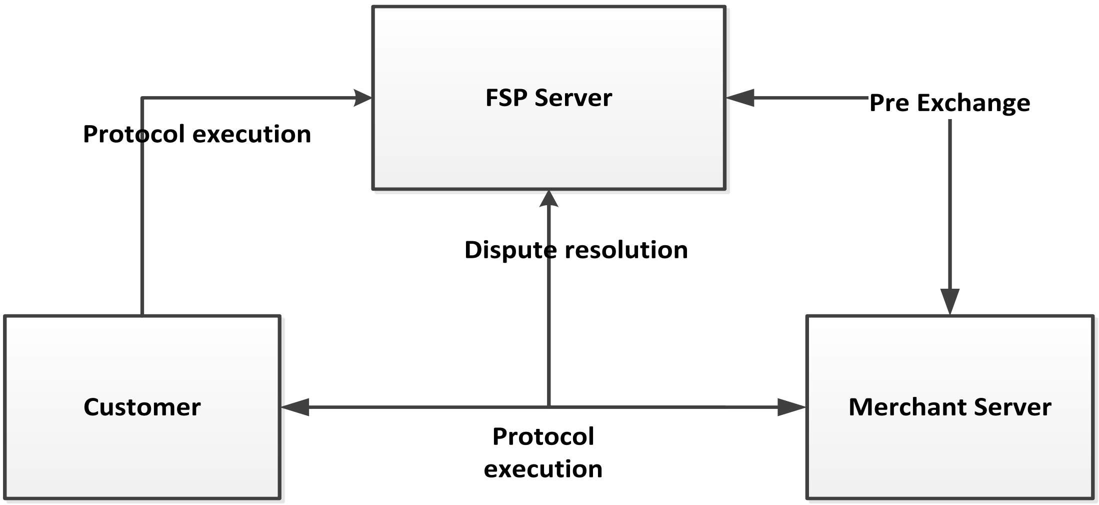


Figure 28: Protocol high level design

In the pre-exchange phase, the merchant server communicates with the FSP server. The merchant then encodes the item using a temporary session key that will be used by the customer to decode the commodity. An invoice is then produced for the purchase of the commodity. The various entities that take part in the exchange phase are the Merchant server, the FSP server and the Customer. During this phase, the protocol messages are exchanged between these parties. The customer communicates with the FSP server in case of any complaint during the exchange phase. The FSP server then confirms the validity of the complaint. If the complaint is genuine, the FSP initiates an automatic resolution for both the Merchant server and the Customer.

6.1.1. Activity diagrams for protocol

Figures 29 and 31 illustrate all the activities that occur during the exchange phase between the Customer, the Merchant server and the FSP server. Figure 30 illustrates the activities of the dispute resolution phase. The specifications of the protocol in Chapter 5 have been used to draw these activity diagrams

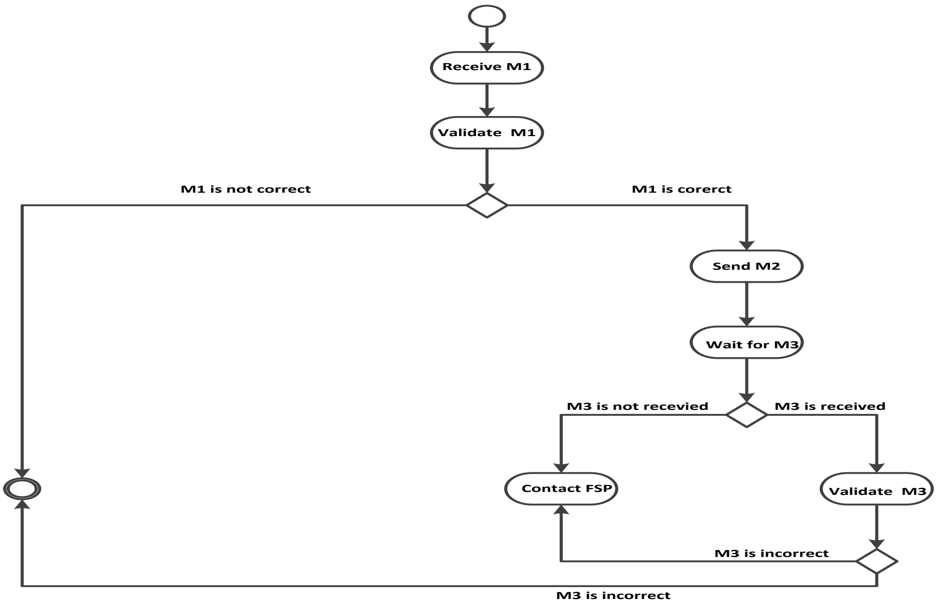


Figure 29: The Customer activity diagram.

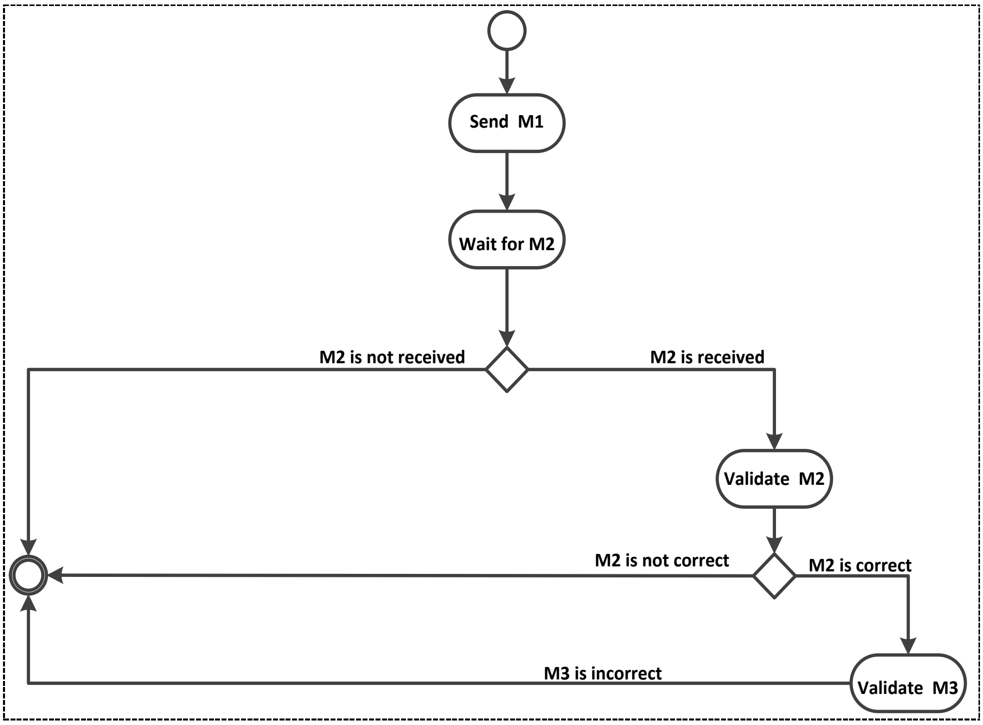


Figure 30: The Merchant server activity diagram.

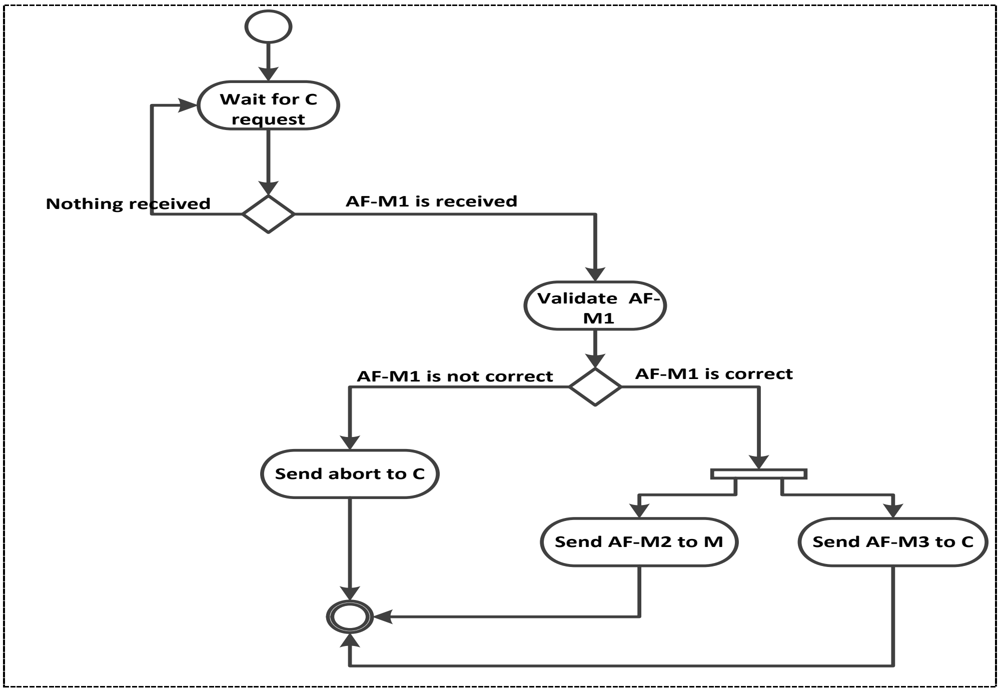


Figure 31: FSP dispute.

6.2. Implementation

A proof of concept (POC) has been developed to provide the basic flows discussed in the proposed protocol. The POC is intended to display the efficiency and enforcement of the proposed protocol in expression of communication cost and speed, and its ability to provide enough information for conflict resolution.

The prototype consists of three modules:

1- ***Merchant:*** this is a Java EE application that is hosted on a Tomcat application server. This module consists of the following major components:

   a. ***RequestHanlder servlet:*** this servlet is responsible for handling the request of the customer and providing the proper response. For example, when the customer selects a certain product, a customer request is stored on the database for further processing.

   b. ***FSPConnector:*** this is a servlet that is responsible for communicating the customer's request to the FSP, sending the details necessary for the FSP to proceed with its obligations. All the communication between the FSP and the Merchant takes place using the SOAP over HTTP protocol. SOAP (Simple Object Access Protocol) is the *de facto* standard used for Web services; it is an XML set of communication standards that are used for the loosely coupled integration of different systems built using different technologies.

   c. ***Product Delivery Handler:*** this component is triggered when a payment confirmation is received from the FSP. This component encrypts the product, compresses it and delivers it to the customer via email or through the web page, depending on the file size.

2- ***FSP*** is another web application developed using Java EE technology, and it too runs on a Tomcat application server. The main components of this module are:

    a. ***MerchantHandler:*** this servlet is responsible for obtaining the payment details sent by the Merchant.

    b. ***CustomerPaymentHandler:*** this is responsible for providing the Customer with the proper interface to obtain the payment details and approval to proceed with acquiring the product.

    c. ***PaymentUpdater:*** this is a scheduled task that uses Quartz for Java. This task monitors any updates on the payment details in order to notify the Merchant.

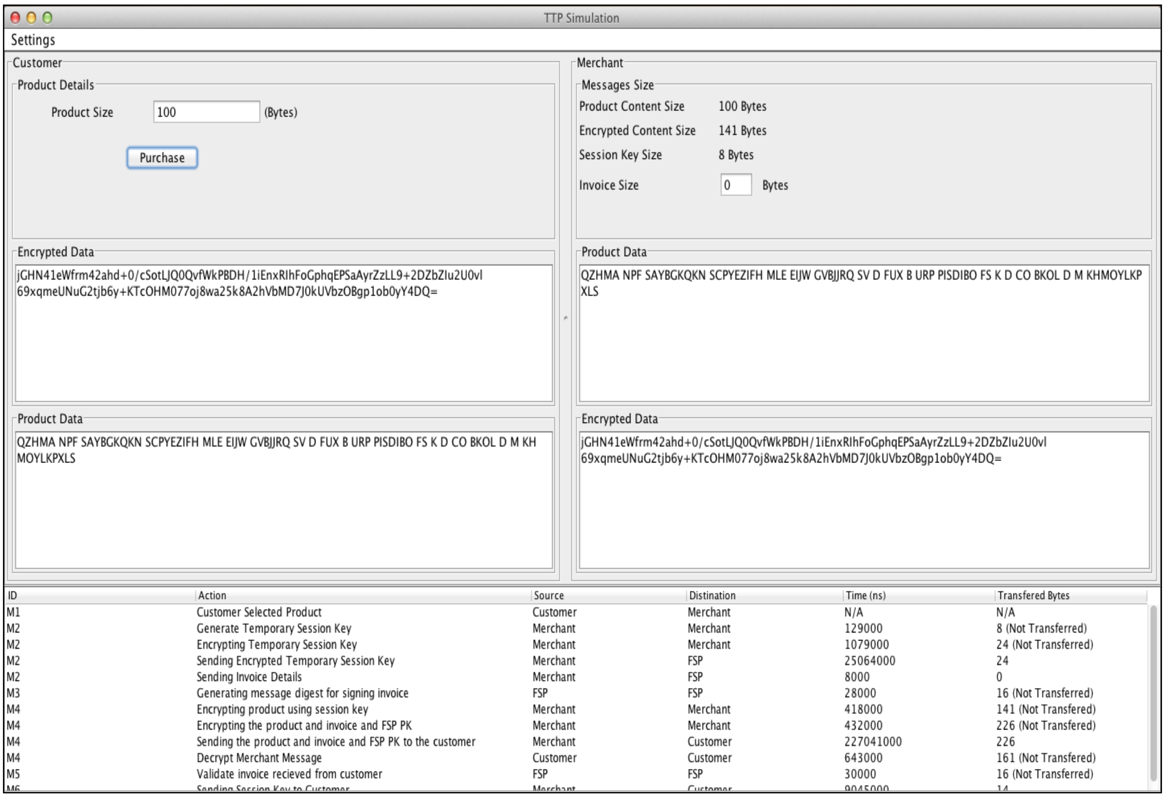
3- ***Customer:*** this is the user of the system, who accesses the system and obtains the information over his browser. connection among the Consumer and the seller and the between the Consumer and the FSP takes place over digitally signed HTTP post requests

6.3. Application Control Panel

In order to generate real readings that can provide accurate measures of the performance of the proposed protocol, a centralized control panel is provided that lists all the communications taking place between the three modules. These communication items are recorded together with some necessary information in order to provide a measurement for the performance of the proposed algorithm. These extra information data include the following:

- Request initiation time
- Request fulfilment time
- Time needed to obtain the response to a call (if any)
- The size of the data being sent over the network

Besides providing the details of the communication between the different parties, the control panel allows for simulating the full cycle of the communication between the different modules, independent of the underlying network infrastructure.



Product Size 100 (Bytes)

Product Content Size 100 Bytes
Encrypted Content Size 141 Bytes
Session Key Size 8 Bytes
Invoice Size 0 Bytes

Encrypted Data

```
jGHN41eWfrm42ahd+0/cSotLJQ0QvfWkPBDH/1iEnxRIhFoGphqEPSaAyrZzLL9+2DZbZIu2U0vl
69xqmeUNuG2tjb6y+KTcOHM077oj8wa25k8A2hVbMD7J0kUVbzOBgp1ob0yY4DQ=
```

Product Data

```
QZHMA NPF SAYBGKQKN SCPYEZIFH MLE EIJW GVBJJRQ SV D FUX B URP PISDIBO FS K D CO BKOL D M KHMOYLKP
XLS
```

Product Data

```
QZHMA NPF SAYBGKQKN SCPYEZIFH MLE EIJW GVBJJRQ SV D FUX B URP PISDIBO FS K D CO BKOL D M KH
MOYLKPXLS
```

Encrypted Data

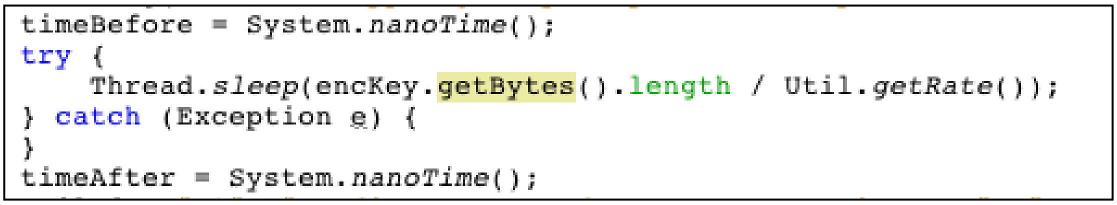
```
jGHN41eWfrm42ahd+0/cSotLJQ0QvfWkPBDH/1iEnxRIhFoGphqEPSaAyrZzLL9+2DZbZIu2U0vl
69xqmeUNuG2tjb6y+KTcOHM077oj8wa25k8A2hVbMD7J0kUVbzOBgp1ob0yY4DQ=
```

ID	Action	Source	Distination	Time (ns)	Transfered Bytes
M1	Customer Selected Product	Customer	Merchant	N/A	N/A
M2	Generate Temporary Session Key	Merchant	Merchant	129000	8 (Not Transferred)
M2	Encrypting Temporary Session Key	Merchant	Merchant	1079000	24 (Not Transferred)
M2	Sending Encrypted Temporary Session Key	Merchant	FSP	25064000	24
M2	Sending Invoice Details	Merchant	FSP	8000	0
M3	Generating message digest for signing invoice	FSP	FSP	28000	16 (Not Transferred)
M4	Encrypting product using session key	Merchant	Merchant	418000	141 (Not Transfered)
M4	Encrypting the product and invoice and FSP PK	Merchant	Merchant	432000	226 (Not Transfered)
M4	Sending the product and invoice and FSP PK to the customer	Merchant	Customer	227041000	226
M4	Decrypt Merchant Message	Customer	Customer	643000	161 (Not Transfered)
M5	Validate invoice recieved from customer	FSP	FSP	30000	16 (Not Transferred)
M6	Sending Session Key to Customer	Merchant	Customer	9045000	14

Figure 32: Application interface

The simulation of the proposed protocol is implemented using Java SE. The user interface in Figure 32 shows the main flow of the protocol simulation, which is the same as the real one. All the communicated data are encrypted using the Java Cryptography APIs.

In order to derive the most accurate measures for the performance of the protocol (using the simulation), no real network communication takes place; instead, the network bandwidth (data rate) is simulated as demonstrated in the Figure below. Providing the measurement in a controlled environment should give more precise results.



```
timeBefore = System.nanoTime();
try {
    Thread.sleep(encKey.getBytes().length / Util.getRate());
} catch (Exception e) {
}
timeAfter = System.nanoTime();
```

Figure 33: Obtaining the time needed for an operation

In this code snippet, the application is forced to wait for a period that is equivalent to the period requirement to transfer the data of size (encKey.getBytes().length) over a network of data rate that is equal to Util.getRate(). Using a real network communication would have resulted in an overhead that would lead to a measurement of not only the application logic itself, but also the communication overhead.

As can also be noticed in the code snippet, the performance is measured using the System.nanoTime() API call; this API is documented in Sun's Java API docs

Another option that is available for measuring the time needed to complete an operation depends on reading the system's millisecond records. This can be done in Java using the currentTimeMillis API call, which is documented in Java docs as well:

System.currentTimeMillis returns the *'wall clock'* time of the computer. The control of this is outside the JVM and can change unpredictably, for example, because of periodic setting of the system time by an ntp service. This means that intervals measured using System.currentTimeMillis will include any changes to the computer's clock.

The method System.nanoTime was introduced in Java 1.5 to address this problem. Whenever an interval is computed (and the JVM is not restarted between start and stop), System.nanoTime should be used instead (converting the result to millis as needed). This is why the use of System.nanoTime should provide a more precise result in terms of performance measurement.

6.4. Security

In the implementation of the suggested protocol, securing the messages between the different parties (Merchant, Customer and FSP) is of paramount importance, and this is why the implementation contains different levels of encryption, as will be detailed later.

The encryption algorithms used in the implementation are done using Java Cryptography APIs and Java Cryptography Extension. The following points detail the encryption algorithm used for each message, and the sources and destinations related to each message:

1- Communication between Merchant and FSP: the encryption is done using the DES algorithm, and the implementation uses an Asymmetric key, which is agreed upon between the Merchant and the FSP prior to starting to exchange data. For making the implementation straightforward, the key pair is randomly created at each application start-up.

2- Communication between the Merchant and the Customer: this is done using DES with the Asymmetric key pair

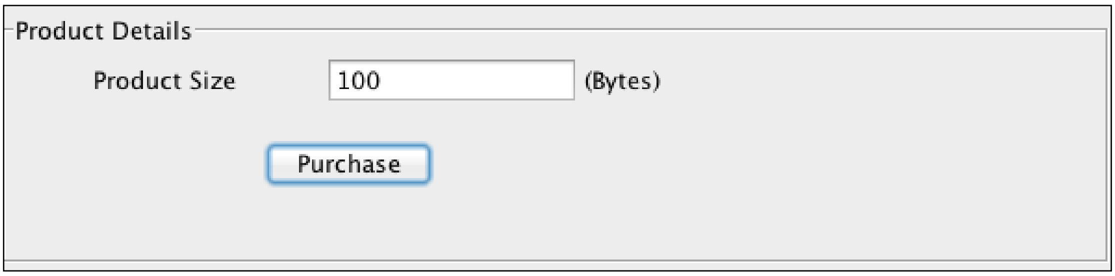
3- Product encryption: the product is encrypted using the temporary Asymmetric key pair, which is exchanged with the Customer and the FSP. The key itself is encrypted using DES.

4- Invoice authenticity check: the invoice that is being sent by the Merchant to the FSP is stored by the FSP in a message digest format. Later, when the buyer gains the invoice from the seller, he will send it to the FSP for validation.

6.5. *User Interface Elements*

- ***Customer Product Selection***

Using this part of the application, the user is able to enter the product size. This approach is used to enable the easy implementation of the system; instead of focusing on boilerplate code, the focus is on implementing the core functionality of the implementation.



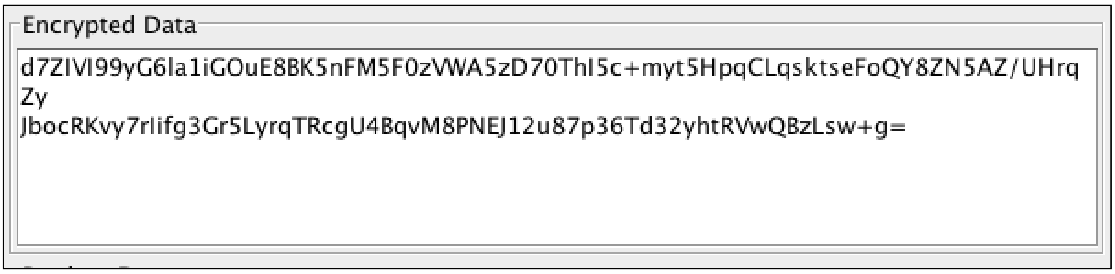
Product Details

Product Size: 100 (Bytes)

Figure 34: Product selection

- ***Encrypted Product Data***

Once the Customer has entered the desired product (simulated by entering the product size), the Merchant generates a random set of data that will resemble the product itself. The product is then encrypted using the session key, and the encrypted data are displayed in the Merchant panel, which are then sent to the Customer and viewed in the Customer panel.

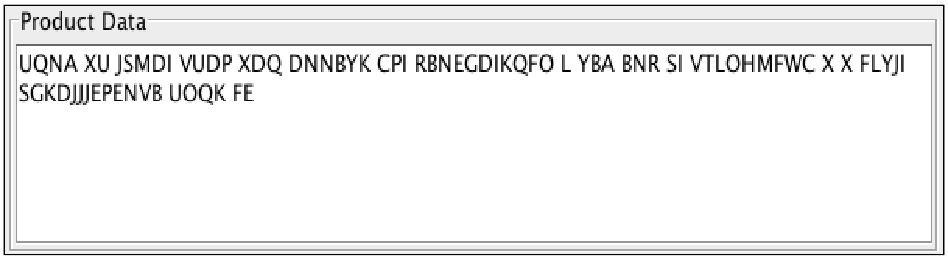


```
d7ZIVI99yG6la1iGOuE8BK5nFM5F0zVWA5zD70ThI5c+myt5HpqCLqsktseFoQY8ZN5AZ/UHrq
Zy
JbocRKvy7rlifg3Gr5LyrqTRcgU4BqvM8PNEJ12u87p36Td32yhtRVwQBzLsw+g=
```

Figure 35: Product data sent to the customer (encrypted)

- ***Product Data***

This panel shows the original product data, which are randomly generated, and to make these data more readable, every couple of characters are randomly separated with a space.

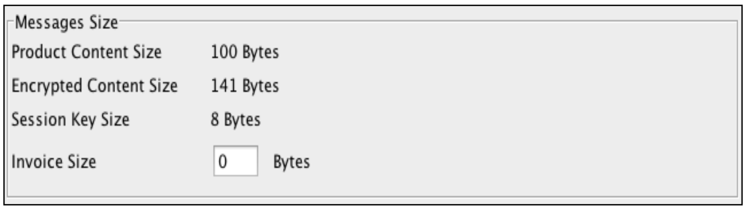


```
UQNA XU JSMDI VUDP XDQ DNNBYK CPI RBNEGDIKQFO L YBA BNR SI VTLOHMFWC X X FLYJI
SGKDJJJEPENVB UOQK FE
```

Figure 36: Original product data (randomly generated)

- ***Merchant Message Details***

This panel shows the sizes of the generated messages, namely, the product data size, the encrypted product data size and the session key size. The last part of the panel is used to simulate real-life messaging, where each request sent to the FSP will contain an invoice. The invoice size can be entered using this field to simulate different invoice data sizes.



Messages Size

Product Content Size 100 Bytes

Encrypted Content Size 141 Bytes

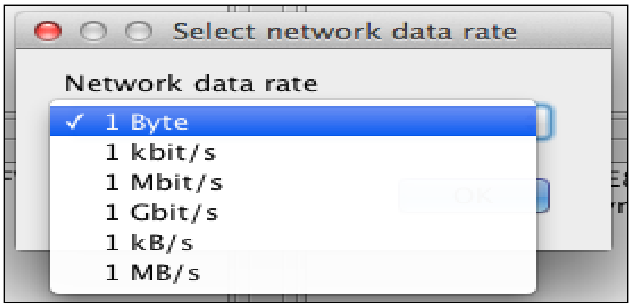
Session Key Size 8 Bytes

Invoice Size 0 Bytes

Figure 37: Messaging details

- ***Network Data Rate Settings***

As discussed earlier, having a controlled environment to operate an application for measurement can be difficult using real communication media because for various reasons. In order to make the results as accurate as possible, network data rates are simulated through a programmed delay that eliminates other factors.



Select network data rate

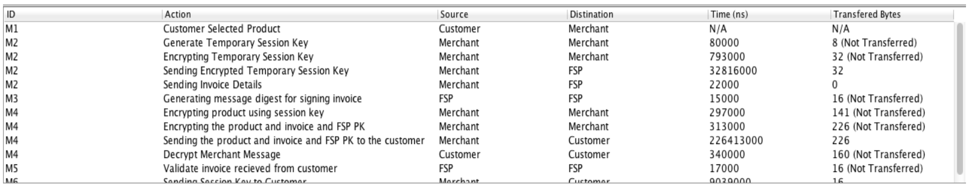
Network data rate

✓ 1 Byte
1 kbit/s
1 Mbit/s
1 Gbit/s
1 kB/s
1 MB/s

Figure 38: Setting the network data rate

- ***Function Analysis***

Every function called during the execution of each operation is detailed in this table. The details include the operation (message ID), the initiator, the receiver, the time needed to complete the function and the data size in bytes.



ID	Action	Source	Distination	Time (ns)	Transfered Bytes
M1	Customer Selected Product	Customer	Merchant	N/A	N/A
M2	Generate Temporary Session Key	Merchant	Merchant	80000	8 (Not Transferred)
M2	Encrypting Temporary Session Key	Merchant	Merchant	793000	32 (Not Transferred)
M2	Sending Encrypted Temporary Session Key	Merchant	FSP	32816000	32
M2	Sending Invoice Details	Merchant	FSP	22000	0
M3	Generating message digest for signing invoice	FSP	FSP	15000	16 (Not Transferred)
M4	Encrypting product using session key	Merchant	Merchant	297000	141 (Not Transfered)
M4	Encrypting the product and invoice and FSP PK	Merchant	Merchant	313000	226 (Not Transfered)
M4	Sending the product and invoice and FSP PK to the customer	Merchant	Customer	226413000	226
M4	Decrypt Merchant Message	Customer	Customer	340000	160 (Not Transfered)
M5	Validate invoice recieved from customer	FSP	FSP	17000	16 (Not Transferred)
M6	Sending Session Key to Customer	Merchant	Customer	9039000	16

Figure 39: Function analysis

- ***Application Settings***

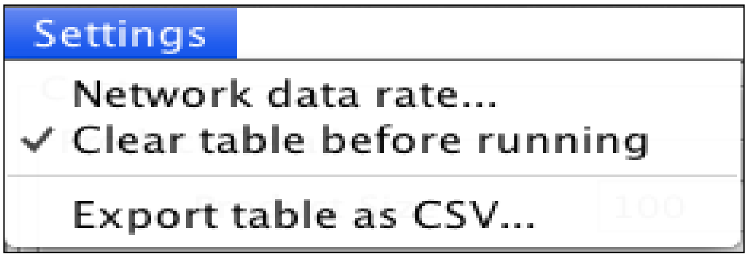


Figure 40: Application Settings

- ***Main Flow Activity Diagrams***

This section demonstrates the three main activities described in the suggested algorithm: Customer-Merchant interaction to select the product and start the purchase process, and the Customer-FSP-Merchant cycle to fulfil the purchase process.

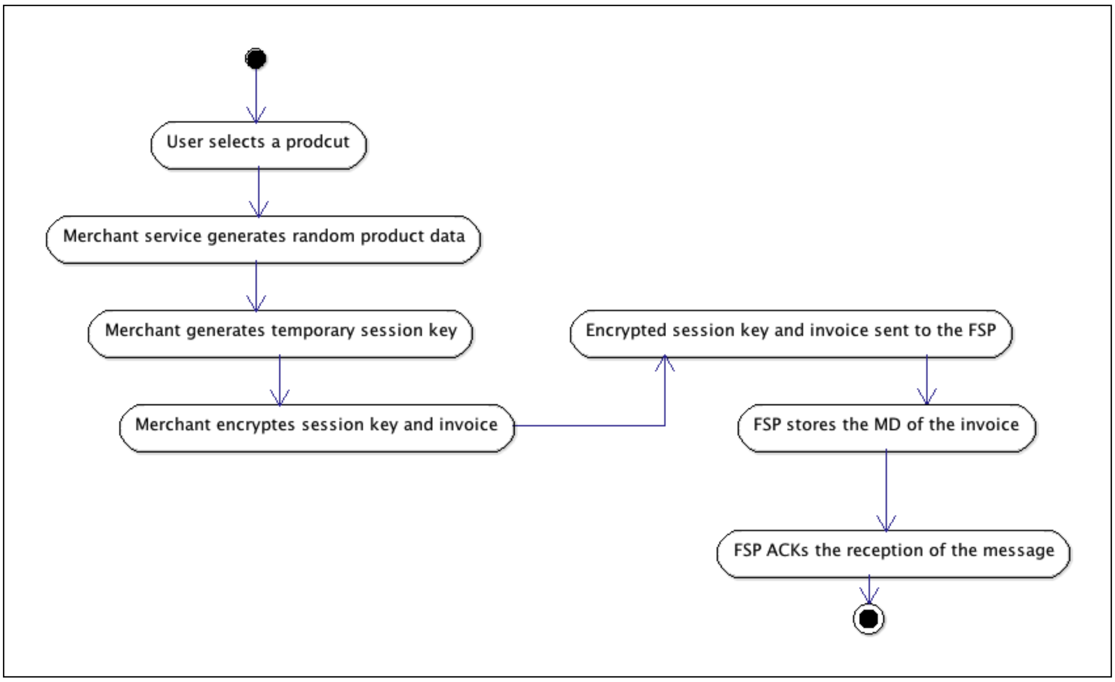

User selects a prodcut
Merchant service generates random product data
Merchant generates temporary session key
Merchant encryptes session key and invoice
Encrypted session key and invoice sent to the FSP
FSP stores the MD of the invoice
FSP ACKs the reception of the message

Figure 41: Product request interaction flow

When the user selects a product, which in the case of this POC is done by entering a product size, the Merchant service receives the Customer's request and then randomly generates product data matching the selected size. The Merchant service then creates the tsk that will be used after while by the FSP for conflict resolution. The temporary session key, product details and invoice details are sent to the FSP after being encrypted using the Merchant's secret key. The FSP service, about to gain the encrypted order from the seller, decrypts the message to extract the details; the FSP then generates a message digest that is stored at the FSP until completion. The message digest is used by means of the buyer to check that the invoice gained from the seller service is authentic. The FSP service finally sends an ACK to the Merchant service, stating that the transaction was successful.

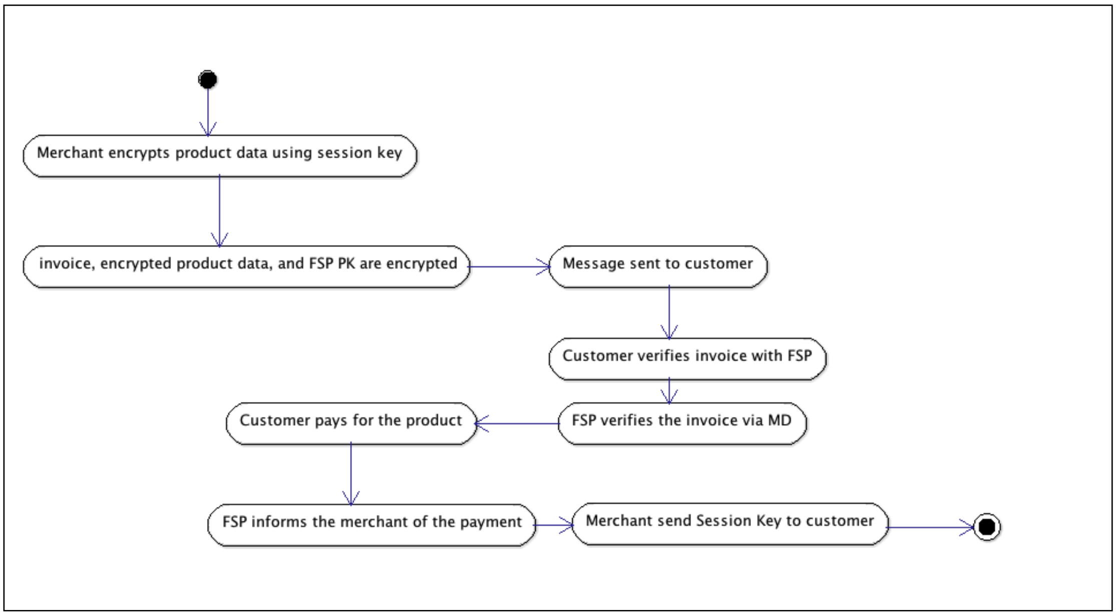

Merchant encrypts product data using session key
invoice, encrypted product data, and FSP PK are encrypted
Message sent to customer
Customer verifies invoice with FSP
FSP verifies the invoice via MD
Customer pays for the product
FSP informs the merchant of the payment
Merchant send Session Key to customer

Figure 42: Product delivery interaction flow

Once the ACK is received from the FSP, the Merchant service encrypts the product by the sk that was rationed together with the FSP. The encrypted product, invoice, and FSP authentication details are encrypted and sent to the Customer. The Customer verifies that the message gained from the Merchant is creditable by sending the request to the FSP service, which checks the request's authenticity against the message digest that was generated and stored in the previous flow. If the request is verified, the buyer is asked to make a payment for the item. The FSP then transmits the payment confirmation to the seller, who in turn transmit the sk to the buyer to be used to decrypt the product.

Finally, if the Merchant fails to send the session key to the Customer, whether deliberately or for technical reasons, the Customer can refer back to the FSP in order to obtain what is rightly his. The flow is detailed in the figure below; the Customer sends the invoice number to the FSP, who checks the validity of the request and that the payment has been fulfilled. The FSP then sends a request to the Merchant with the invoice number, asking the Merchant to correct the status of the related product request. The FSP will then wait for a pre-set timeout and then checks back with the Customer to see if the Merchant has sent the session key. If not, the FSP sends the stored session key to the Customer. The same scenario can be triggered if the Merchant sends an incorrect session key and the Customer is unable to decrypt the product.

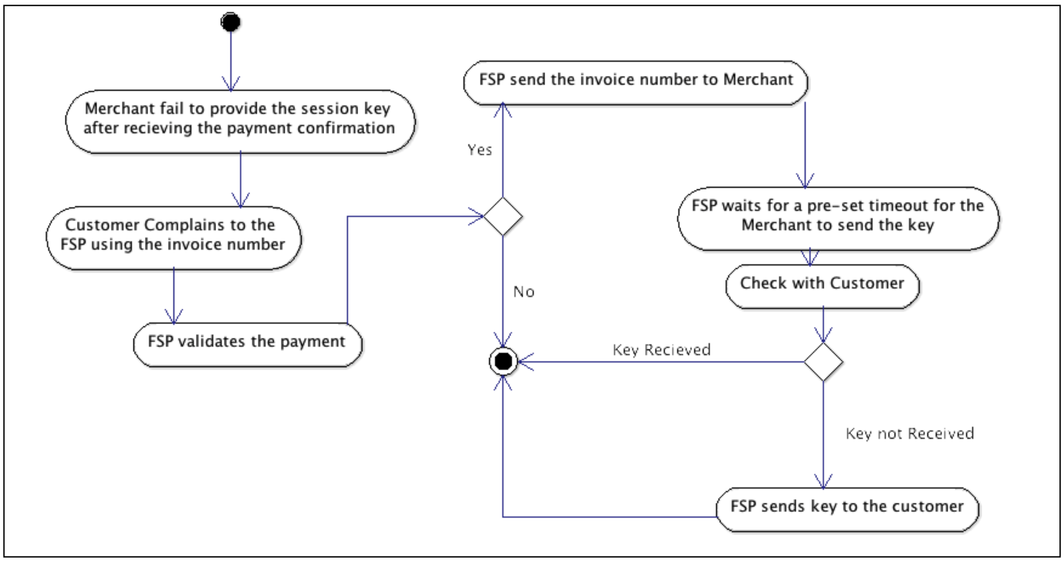

Merchant fail to provide the session key after recieving the payment confirmation
Customer Complains to the FSP using the invoice number
FSP validates the payment
Yes
No
FSP send the invoice number to Merchant
FSP waits for a pre-set timeout for the Merchant to send the key
Check with Customer
Key Recieved
Key not Received
FSP sends key to the customer

Figure 43: Conflict resolution flow

6.6. Performance Evaluations

To measure the performance of the proposed protocol, we performed a emulation of the Zhang *et al*. protocol [43] and the Devane *et al*. protocol [44], and compared the results of each one with our protocol. The purpose of performance evaluations is to measure the computation time of each message, and to determine the total computation time. The compration Properties are the time for constructing the messages, the time for generates and verify the messages and the total time for send the message.

ID	Action	Scenario 1: digital product size = 28KB	Scenario 2: digital product size = 2.2MB
Pr-M1	*Generating temporary session key*	14	35
Pr-M1	*Encrypting temporary session key*	12	38
Pr-M1	*Sending encrypted temporary session key*	23.7	239
Pr-M1	*Sending invoice details*	4	4
Pr-M2	*Generating message digest for signed invoice*	3	29
M1	*Encrypting product using session key*	34	53.6
M1	*Encrypting the product, invoice and FSP PK*	20	98
M1	*Sending the product, invoice and FSP PK to the customer*	147	537
M1	*Decrypting the Merchant message*	47	59
M2	*Validating the invoice received from the Customer*	3	3
M3	*Sending the session key to the Customer*	61	78
	Total	***368.7***	***1173.6***

Table 11: Timing of the proposed protocol

ID	Scenario 1: digital product size = 28KB	Scenario2:digital product size = 2.2MB
Message 1 construction	34	49
Message 1 verification	24	47
Message 2 construction	56	406
Message 2 verification	26.2	67
Message 3 construction	61.2	94
Message 4 construction	67	77
Payment decryption	186.7	211
Digital product decryption	188.8	671
Total	**643.9**	***1622***

Table 12: Timing of the Zhang *et al*. protocol

ID	Scenario 1: digital product size = 28KB	Scenario 2: digital product size = 2.2MB
Message 1 construction	54	67
Message 1 verification	34	77
Message 2 construction	65	493
Message 2 verification	44	78
Message 3 construction	65	94
Message 4 construction	67	76
Payment decryption	193	271
Digital product decryption	199	701
Total	**721**	**1857**

Table 13: Timing of the Devane *et al*. Protocol

The comparison outcome display that the new protocol is more efficacious than either the Zhang *et al*. protocol [43] or the Devane *et al*. protocol [44].

By testing our protocol prototype, our work has proved that the process of the proposed protocol is workable. When it is implemented in the real world, the length of the asymmetric keys becomes the main factor affecting the protocol’s performance; in other words, increasing the length of the asymmetric keys (to generate the highest possible level of security) will increase the computation time significantly.

6.7. Summary

The design and implementation of the prototype presented in this chapter. Also the tools that have been implemented to easily execute all the scenarios of the protocols presented.

Chapter 7

Conclusions and Future Work

Objectives:

- ***Summary of thesis***
- ***Measure of Success***
- ***Future Works***

7.1. Summary of Thesis

Due to the rapid growth of electronic commerce in recent years, many businesses are today conducted online. In other words, more businesses than ever before are using the Internet to sell their commodities to people all over the world. The internet provides them with a platform for selling their items to all kinds of people without the restrictions of geographical borders. Customer choice in buying goods and services has been greatly improved through this growth in e-commerce, and for a variety of reasons, growing numbers of customers now opt to buy their items through the Internet [112,113]. Firstly, customers have the convenience of making purchases in the comfort of their homes without having to go to shopping malls or having to suffer the hassle of traffic jams. Secondly, customers have the opportunity to quickly compare the prices of various traders. Thirdly, goods and services are delivered to the customer’s home. Lastly, customers are able to buy products at anytime from anywhere in the world.

Accordingly, there needs to be a system in place for ensuring that all data being sent is done so through secure means. There is no doubt that e-commerce has made the exchange of goods and services easier but it also poses risks to both the customer and the merchant, including the issues of security, safeguarding users’ privacy, trust and anonymity. The main challenge lies in the exchange of the digital commodities among the Participants ingaged in the operation. This is because,no previous dealing among the transacting Participants so they maybe no confidence amont them. For overcome this, the concept of fair exchange has been developed, and this manifested through protocols that underpin the transaction process [112,113].

Fair exchange protocols are divided into two: the ones that involve the participation of a Trusted Third Party (TTP), and the ones that do not. The TTP guarantees fair exchange of products among the transacting Participants. The protocols do not use trusted third party use the idea of submitting the product in parts. For instance, the customer transmits part of the payment while the merchant transmits part of the item. The exchange process goes on until the complete item is traded.

The protocols that use a TTP are containin different kinds: the first is called an inline TTP-based protocol. The role of the TTP is to exchange the respective commodities. In other words, each entity submits its product to TTP, and the TTP then delvers the digital commodity to the buyer and the payment to the merchant. In this case, the TTP guarantees that fairness is achieved in the protocol [112,113].

The second type of protocol uses an online TTP. The role of the online TTP is only to verify the items to be exchanged, and hence the participation of the trusted third party in this protocol is reduced. The third kind of protocol is offline . In these protocols, the transacting entities trade each party product directly, and the trusted third party only participates to solve problems; hence, its involvement in the protocol is minimized.

Above, we have discussed the two kinds of fair exchange protocol (those that use a TTP and those that do not). Their main aim is to guarantee the fair exchange of products for the transacting parties. Protocols that do not use a TTP are much less effective because the process required to complete them is long; in order to obtain fairness for the transacting entities, the protocols should have equal computational power. On the other hand, the major limitation of the inline protocols use the trusted third party is that the trusted third party can become the source of a bottleneck but it should always be present. Also, the protocols cannot submit the products to the transacting parties if the TTP crashes [112,113].

When we compare protocols that use an online TTP and the ones that involve the use of an inline TTP, we realise that they both have setbacks. The difference is that in online TTP, the participation of the TTP is greatly reduced. Generally speaking, in the offline protocols, the TTP is not engaged in the protocols unless there is a conflict between the transacting parties.

This thesis has focused on online fair exchange protocols to exchange payments and items between customers and merchants. The work reported in this thesis first studied the existing fair exchange protocols that seek to address the fairness problem. Then, we proposed a more efficient protocol to serve this purpose. The original idea in this thesis is to reduce the communication overheads, risks and solve the bottleneck problems in the protocols that involve an online TTP. The idea is to dividing the process to two phases, pre exchange phase and exchange phase. The proposed protocol has the characteristics: three messages are required between all parties, the protocol guarantee strong fairness for both customer and merchant. The new protocol let the customer to be sure about the merchant's item before he send his item and let the merchant to be sure about the customer's item before he send his item, online disputes are resolved by a Financial Service Provider (FSP) [112,113].

7.2. Measure of Success

The results reported in this thesis began with a set of aims labelled the Measure of Success; these were enumerated in Chapter 1. This section addresses each of the criteria to ascertain the degree to which the research has been successful.

The proposed protocol has the following features: only three messages are required to be exchanged among all the parties involved; this is the lowest number of messages when compared with the other relevant fair exchange protocols in the literature. With regard to the number of messages in the dispute resolution phase, the lowest possible number of messages to be executed is three. Therefore, the number of messages exchanged during the proposed protocol (in both the exchange and dispute phases) is kept to a minimum.

The protocol guarantees strong fairness for both customer and merchant. IT allows both parties (customer and merchant) to check the correctness of the item of the other party before they send their item; disputes are resolved automatically online by a Financial Service Provider (FSP), which is efficient in that it has a low number of modular exponentiations.

The SPIN model checker is the system that was used to verify the fairness of our protocol. When using the SPIN model checker for the authentication of the fairness property of the protocol, the fairness property was specified using LTL. All the participants in the protocol are represented by a process in Promela, and their activities are also specified in the process.

A proof of concept (POC) was developed to describe the basic flows within the proposed protocol. The POC is intended to show the efficiency and assess the performance of the proposed protocol in terms of communication cost and speed, and its ability to provide enough information for conflict resolution.

The comparisons will be made in accordance with a number of different properties; (1) How many massages required in every phase?, (2) Is the trusted third party save the item?, (3) Which party will arise the dispute?, and (4)compare the modular exponentiations numbers .These different criteria will help improve the protocol by reduce the time for send and receive the messages, Reduce the time for generates and verify the messages and the use of TTP is only for verifying the items not use for delivering or store the items that will be exchange will be reduce the load on network.

7.4. Future Works

The proposed e-commerce protocol was completed in the time available. However, some valuable researches remain to be done in the future.

- This researcher plans to modify the protection method for digital contents through which the merchant can use different symmetric keys to encrypt a specific digital content for different customers. This modification would provide the merchant with a higher degree of security for digital-content protection, and would also prevent collusion between customers (in trying to obtain the decryption key of a specific digital content).

- Some customers prefer to remain anonymous (with respect to merchants). The protocols presented in this thesis do not offer such anonymity to the customer; this is because the merchant can identify the customer from the payment. Future work could study how to provide anonymity for customers within the proposed protocol.

- Investigate the use of more than one TTP in the protocol.

- The parties involved in the exchange phase of the proposed protocol are C and M; i.e. there are two parties. Future work could extend the ideas of the proposed protocol by generating multi-party fair exchange protocols.

- Besides the RSA-based electronic cash, this researcher intends to consider other payment methods (under the customer-anonymity characteristic) to improve the computational efficiency of the protocol.

- The protocol presented in this thesis is for the exchange of a payment and a items. Future work could extend this so that our protocol could be used for physical products.
- Existing fault-tolerant techniques from the literature should be combined with the proposed protocol in order to generate a fully fault-tolerant fair exchange protocol. This will guarantee fairness for all Participants even in the event of a system crash and/or communication failure.

References

[1] J. Zhou and D. Gollman, "A fair non-repudiation protocol," 1996, pp. 55-61

[2] W. Wang et al, "Model Checking - a rigorous and efficient tool for e-commerce internal control and assurance", Emory University, USA, 2001

[3] Electronic commerce, (2010, January 24), [online], available, http://en.wikipedia.org/wiki/Electronic_commerce.

[4] A. Alaraj, "Enforcing Honesty in E-Commerce Fair Exchange Protocols," Durham University, 2008.

[5] D. Whiteley, E-commerce: Strategy, technologies and applications: McGraw-Hill Publishing Company, 2000.

[6] Wordpress, (2010, novmber), [online], available, http://darojorojo5.files.wordpress.com/2010/11/ecomsteps.jpg

[7] Z. Zhang, "Some Issues about E-commerce in the WTO Framework with Implications for China's Laws," World Trade Institute (WTI), 2004

[8] D. C. Lynch and L. Lundquist, Digital money: the new era of Internet commerce: John Wiley & Sons, Inc., 1996.

[9] G. Medvinsky and C. Neuman, "NetCash: A design for practical electronic currency on the Internet," 1993, pp. 102-106.

[10] D. Abrazhevich, "Classification and characteristics of electronic payment systems," Electronic Commerce and Web Technologies, pp. 81-90, 2001

[11] S. Subramanya and B. K. Yi, "Digital rights management," Potentials, IEEE, vol. 25, pp. 31-34, 2006

[12] N. Asokan, et al., "The state of the art in electronic payment systems," Computer, vol. 30, pp. 28-35, 1997

[13] E. Turban, et al., "Electronic Commerce 2004: A Managerial Perspective, 2004," ed: Prentice Hall.

[14] W. Ford and M. S. Baum, Secure electronic commerce: building the infrastructure for digital signatures and encryption: Prentice Hall PTR, 2000.

[15] BBC NEWS, (2010, MAY 20), [online], available, http://www.bbc.co.uk/news/business/.

[16] C. Ruppel, et al., "E-commerce: the roles of trust, security, and type of e-commerce involvement," E-service Journal, vol. 2, pp. 25-45, 2003.

[17] D. Cook and W. Luo, "The role of third-party seals in building trust online," E-service Journal, vol. 2, pp. 71-84, 2003.

[18] B. Violino, "Building B2B trust," Computerworld, vol. 36, pp. 32-33, 2002.

[19] T. Tsiakis and G. Sthephanides, "The concept of security and trust in electronic payments," Computers & Security, vol. 24, pp. 10-15, 2005.

[20] B. J. Corbitt, et al., "Trust and e-commerce: a study of consumer perceptions," Electronic commerce research and applications, vol. 2, pp. 203-215, 2003.

[21] S. Srinivasan, "Role of trust in e-business success," Information management & computer security, vol. 12, pp. 66-72, 2004.

[22] eBay,(2010, JANE 12), [online], available, http://www.ebay.com/.

[23] F. C. Gartner, et al., "Approaching a formal definition of fairness in electronic commerce," 1999, pp. 354-359.

[24] The free dictionary, [online], available, http://www.thefreedictionary.com/fairness.

[25] Advanced services architect.(1999), [online], Available http://www.semper.org/deliver/d10/d10.ps.gz.

[26] N. Asokan,"Fairness in electronic commerce", University of Waterloo,1998.

[27] H. Bürk and A. Pfitzmann, "Value exchange systems enabling security and unobservability," Computers & Security, vol. 9, pp. 715-721, 1990.

[28] A. Nenadi and N. Zhang, "Non-repudiation and Fairness in Electronic Data Exchange," Enterprise Information Systems V, pp. 286-293, 2005.

[29] I. Ray and N. Natarajan, "An anonymous and failure resilient fair-exchange e-commerce protocol," Decision Support Systems, vol. 39, pp. 267-292, 2005

[30] M. Schunter, "Optimistic fair exchange," Universitätsbibliothek, 2000.

[31] A. Nenadic, "A security solution for fair exchange and non-repudiation in e-commerce," Citeseer, 2005.

[32] S. Kremer, et al., "An intensive survey of fair non-repudiation protocols," Computer Communications, vol. 25, pp. 1606-1621, 2002.

[33] I. Ray and N. Narasimhamurthi, "A fair-exchange e-commerce protocol with automated dispute resolution," Data and Application Security, pp. 27-38, 2002.

[34] J. Zhou and D. Gollmann, "Observations on non-repudiation," 1996, pp. 133-144.

[35] I. Ray and N. Natarajan, "An anonymous and failure resilient fair-exchange e-commerce protocol," Decision Support Systems, vol. 39, pp. 267-292, 2005.

[36] P. Liu, et al., "Avoiding loss of fairness owing to process crashes in fair data exchange protocols," 2000, pp. 631-640.

[37] N. Asokan, et al., "Optimistic protocols for fair exchange," 1997, pp. 7-17.

[38] V. Shmatikov and J. C. Mitchell, "Analysis of a fair exchange protocol," 2000.

[39] S. Micali, "Simple and fast optimistic protocols for fair electronic exchange," 2003, pp. 12-19.

[40] N. Zhang, et al., "Practical and efficient fair document exchange over networks," Journal of network and computer applications, vol. 29, pp. 46-61, 2006.

[41] A. Nenadić, et al., "Fair certified e-mail delivery," 2004, pp. 391-396.

[42] A. Nenadić, et al., "RSA-based Certified Delivery of E-Goods Using Verifiable and Recoverable Signature Encryption1," Journal of Universal Computer Science, vol. 11, pp. 175-192, 2005..

[43] Q. Zhang, et al., "A practical fair-exchange e-payment protocol for anonymous purchase and physical delivery," 2006, pp. 851-858.

[44] S. Devane, et al., "Secure e-commerce protocol for purchase of e-goods-using smart card," 2007, pp. 9-14

[45] Q. Zhang, et al., "A Practical E-Payment Protocol To Realize Fair-Exchange," 2004..

[46] J. Zhou and D. Gollman, "A fair non-repudiation protocol," 1996, pp. 55-61.

[47] M. Blum, "How to exchange (secret) keys," ACM Trans. Comput. Syst, vol. 1, 1983.

[48] S. Even, et al., "A randomized protocol for signing contracts," Commun. ACM, vol. 28, 1985

[49] S. Even, et al., "A randomized protocol for signing contracts," Commun. ACM, vol. 28, 1985

[50] M. Schunter, "Optimistic fair exchange", University of Saarland, 2000.

[51] H. Pagnia, et al., "Fair exchange," The Computer Journal, vol. 46, pp. 55-75, 2003.

[52] J. Hörnle, "Online dispute resolution in business to consumer e-commerce transactions," The Journal of Information, Law and Technology (JILT), vol. 2, pp. 02-2, 2002.

[53] S. Alfuraih and R. Snow, "ODR and the E-commerce. In proceedings of Web Technologies, Applications, and Services", conference 2005, Canada. pp.182-186, 2005.

[54] N. Asokan, et al., "The state of the art in electronic payment systems," Computer, vol. 30, pp. 28-35, 1997.

[55] B. Cox, et al., "NetBill security and transaction protocol," 1995.

[56] S. Devane., "Technical Challenges in Smart Card based Indian Payment System for Limited-Connectivity Environments", indian institute of technology, 2006.

[57] G. J. Holzmann, "Design and Validation of," Computer Protocols, 2007.

[58] J. Blackledge, "Cryptography and Steganography: New Algorithms and Applications," 2011.

[59] A. Salomaa, Public-key cryptography vol. 23: Springer, 1996

[60] Artic soft Technologies (2011, December 14), Introduction to Encryption, [Online]. Available. http://www.articsoft.com/wp_explaining_encryption.htm.

[61] C. Ellison and B. Schneier, "Ten risks of PKI: What you're not being told about public key infrastructure," Comput Secur J, vol. 16, pp. 1-7, 2000.

[62] B. Schneier and P. Sutherland, Applied cryptography: protocols, algorithms, and source code in C: John Wiley & Sons, Inc., 1995

[63] W. Stallings, "Cryptography and network security, principles and practices, 2003," Practice Hall.

[64] R. L. Rivest, et al., "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, pp. 120-126, 1978.

[65] B. B. Anderson, et al., "Standards and verification for fair-exchange and atomicity in e-commerce transactions," Information Sciences, vol. 176, pp. 1045-1066, 2006.

[66] W. Wang, et al., "E-process design and assurance using model checking," Computer, vol. 33, pp. 48-53, 2000.

[67] B. B. Anderson, et al., "The application of model checking for securing e-commerce transactions," Communications of the ACM, vol. 49, pp. 97-101, 2006.

[68] E. Clarke and E. Emerson, "Design and synthesis of synchronization skeletons using branching time temporal logic," 25 Years of Model Checking, pp. 196-215, 2008.

[69] L. Lamport, "Sometime is sometimes not never: On the temporal logic of programs," 1980, pp. 174-185.

[70] R. Alur, et al., "Alternating-time temporal logic," Journal of the ACM (JACM), vol. 49, pp. 672-713, 2002.

[71] S. Kremer and J. F. Raskin, "A game-based verification of non-repudiation and fair exchange protocols," CONCUR 2001—Concurrency Theory, pp. 551-565, 2001

[72] G. Holzmann and S. M. Checker, "The: Primer and Reference Manual," ed: Addison

Wesley Professional, 2004

[73] A. Nenadic, "A security solution for fair exchange and non-repudiation in e-commerce," Citeseer, 2005.

[74] J. Garcia-Fanjul, et al., "Formal Verification and Simulation of the NetBill Protocol Using SPIN1," 1998, pp. 195-210

[75] E. M. Clarke, et al., Model checking: MIT press, 2000.

[76] I. Ray and H. Zhang, "Experiences in developing a fair-exchange e-commerce protocol using common off-the-shelf components," Electronic commerce research and applications, vol. 7, pp. 247-259, 2008.

[77] SPIN Manual, (2010, October 25) [Online] Available, www.spinroot.com.

[78] I. Ray, "Failure analysis of an e-commerce protocol using model checking," 2000, pp. 176-183

[79] Java, (2012, April 24), [online], availabl, www.java.sun.com/products/javacard.

[80] Microsoft, (2012, April 26), [online], available, www.microsoft.com/technet/security/guidance/identitymanagment/smrtcdcb/sec1/smartc01.mspx.

[81] T. Ahn, et al., "The impact of the online and offline features on the user acceptance of Internet shopping malls," Electronic commerce research and applications, vol. 3, pp. 405-420, 2005.

[82] B. B. Anderson, et al., "The application of model checking for securing e-commerce transactions," Communications of the ACM, vol. 49, pp. 97-101, 2006.

[83] B. B. Anderson, et al., "Standards and verification for fair-exchange and atomicity in e-commerce transactions," Information Sciences, vol. 176, pp. 1045-1066, 2006.

[84] S. Alfuraih and R. Snow, "ODR and the E-commerce," 2005.

[85] A. Alaraj and M. Munro, "An efficient e-commerce fair exchange protocol that encourages customer and merchant to be honest," Computer Safety, Reliability, and Security, pp. 193-206, 2008.

[86] M. Srivatsa, et al., "Exchangeguard: A distributed protocol for electronic fair-exchange," 2005, pp. 105b-105b.

[87] W. Wang, et al., "E-process design and assurance using model checking," Computer, vol. 33, pp. 48-53, 2000.

[88] Q. Zhang, et al., "A mutual authentication enabled fair-exchange and anonymous e-payment protocol," 2006, pp. 20-20.

[89] N. Zhang, et al., "A unified approach to a fair document exchange system," Journal of Systems and Software, vol. 72, pp. 83-96, 2004

[90] N. Zhang, et al., "Practical and efficient fair document exchange over networks," Journal of network and computer applications, vol. 29, pp. 46-61, 2006.

[91] S. F. Tzeng, et al., "A secure on-line software transaction scheme," Computer Standards & Interfaces, vol. 27, pp. 303-312, 2005.

[92] A. R. Sadeghi and M. Schneider, "Electronic payment systems," Digital Rights Management, pp. 113-137, 2003

[93] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," Communications of the ACM, vol. 24, pp. 533-536, 1981.

[94] R. A. Kemmerer, "Analyzing encryption protocols using formal verification techniques," Selected Areas in Communications, IEEE Journal on, vol. 7, pp. 448-457, 1989.

[95] Axalto and Gemplus to combine to create a global leader in Digital Security, (2010, December 07), [online], available, http://www.gemplus.com/press/archives/2005/corporate/07-12-2005.

[96] S. Devane, et al., "Secure e-commerce protocol for purchase of e-goods-using smart card," 2007, pp. 9-14

[97] Needham R., Schroeder M., "Using Encryption for Authentication in large networks of computers", Communications of the ACM, 21(12), 1978, pp. 993-999.

[98] H. Pagnia, et al., “E-COMMERCE SECURITY," Canada: John Wiley. ISBN 0471192236, 1998

[99] B. Galitsky and B. Kovalerchuk, "Analyzing Attitude in Customer Emails: A Tool for Complaint Assessment," 2006.

[100] C. Hsieh, "E-commerce payment systems: critical issues and management strategies," Human Systems Management, vol. 20, pp. 131-138, 2001

[101] R. Jones, "The PayPal Phenomenon: Lessons from the Leading Edge of Online Payments," CommerceNet Security and Internet Payments Research, 2001.

[102] R. S. AR and S. Devane, "Formal Verification of Payment protocol using AVISPA," 2010.

[103] H. Pagnia, et al., "Fair exchange," The Computer Journal, vol. 46, pp. 55-75, 2003.

[104] W. Gao, et al., "An Abuse-Free Optimistic Fair Exchange Protocol Based on BLS Signature," 2008, pp. 278-282.

[105] X. Li, et al., "Analysis of Offline Fair Exchange Protocols in Strand Spaces," 2008, pp. 272-276.

[106] J. Zhou, et al., "Some remarks on a fair exchange protocol," 2000, pp. 46-57.

[107] H. Shen, et al., "Analyzing fair exchange protocols with strand spaces," MINIMICRO SYSTEMS-SHENYANG-, vol. 27, p. 62, 2006.

[108] S. Kremer, “Formal Analysis of Optimistic Fair Exchange Protocols”, Universit'e Libre de Bruxelles, 2004.

[109] B. B. Anderson, et al., "Model checking for E-business control and assurance," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 35, pp. 445-450, 2005.

[110] I. Ray, "Fair exchange in e-commerce," ACM SIGecom Exchanges, vol. 3, pp. 9-17, 2002

[111] M. Wahab, "Online dispute resolution and digital inclusion: Challenging the global digital divide," 2005, p. 2006.

[112] A. AlOtaibi and H. Aldabbas, "A review of fair exchange protocols", International Journal of Computer Networks & Communications (IJCNC), vol.4, No.4, 2012.

[113] A.alotaibi and H.Aldabbas, "Design and Evaluate a fair exchange protocol Based on online TTP", International Journal of Security and Its Applications (IJSIA), Vol.4, No.4, 2012

Appendix A

This appendix includes example of Protocol that is based on an inline TTP

Ray et al [33] suggested a fair exchange protocol which is based on the theory of cross validation [76]. The cross validation theory therefore allows the customer to verify the coded digital product which is expected from the merchant without decrypting it.

This is an inline TTP based protocol that deals with the process of exchanging a digital product (held by a merchant) with a payment (held by a customer). The digital products of the merchant are advertised in the TTP's catalogue. The following is an explanation of the process of advertisement. A key pair K_1 (which is for encryption) and K_1^{-1} (which is for decryption) is generated by the merchant; and then the merchant delivers them to the TTP, that is the key pair together with the digital product.

The digital product is then coded by the TTP with K_1 and then advertised in the catalogue of the TTP together with the specifications (the same thing is done for each digital product that the merchant wants to exchange for a payment i.e. to sell). Anyone who wants to buy digital products can go online and search the TTP's catalogue and if they find any digital products that interest them, then they download from the websit of the trusited third party a coded e-item.

During the execution of this protocol, the transacting parties and the TTP exchange six messages. The process of exchange begins with the customer sending the first message with a purchase order to the merchant. The merchant authenticates the first message from the customer and if satisfied, then the merchant takes the following steps. Another key pairK_2 and K_2^{-1} that is mathematically related to the first pair is made by the merchant (that is sent to the TTP). After that, the seller transfers the second **m** with the signature of the merchant on the purchase order, and K_2^{-1} to the TTP.

The customer then receives from the merchant a third message with the encrypted digital product. Upon receiving the third message, the customer makes a comparison of the coded digital product that was transmitted by the merchant in the third message with the coded e-item that was downloaded. If satisfied with the comparison, the customer transfer the message number four which contains the purchase order, the signature of the customer on the encrypted digital product, and a signed payment to the TTP. After getting the fourth message, the TTP makes a comparison of the hash value of the purchase order that was sent by the customer (in the fourth message) with the hash value of the purchase order that was sent by the merchant (in the second message).

If the TTP is satisfied with the comparison, it communicates with the customer’s banks for verification of the financial payment. After authenticating the validity of the payment, the TTP sends a fifth message and then submits the payment to the merchant in the sixth message. The TTP is contacted in case of any disputes.

In this protocol, it is clear that, despite the delivery of the encrypted digital product by the merchant to the customer in the third message (i.e. there is a direct communication between the customer and the merchant), there is the involvement of the inline TTP in submitting the goods to the concerned entities. Which means the role of the inline TTP in this scenario is to submit (1) the decryption key of the encrypted e-item to the buyer (encrypted e-item was received from the merchant by the customer in message three), and (2) the payment to the merchant.

Appendix B

This appendix includes example of Protocol that is based on an offline TTP

Ray et al[26] proposed an optimistic fair exchange protocol that relies on the notion of the theory of cross validation [76]. Before the protocol is initiated, there are some steps which are carried out. A merchant has to enroll with TTP and the TTP then creates the key pair. Trusted third party after that gives merchant the KM_1 and retains KM_1^{-1}. On the other hand, a buyer should open account in a bank. Then the bank creates the key pairs KC_1 and KC_1^{-1} and then gives C with KC_1 and retains KC_1^{-1}.

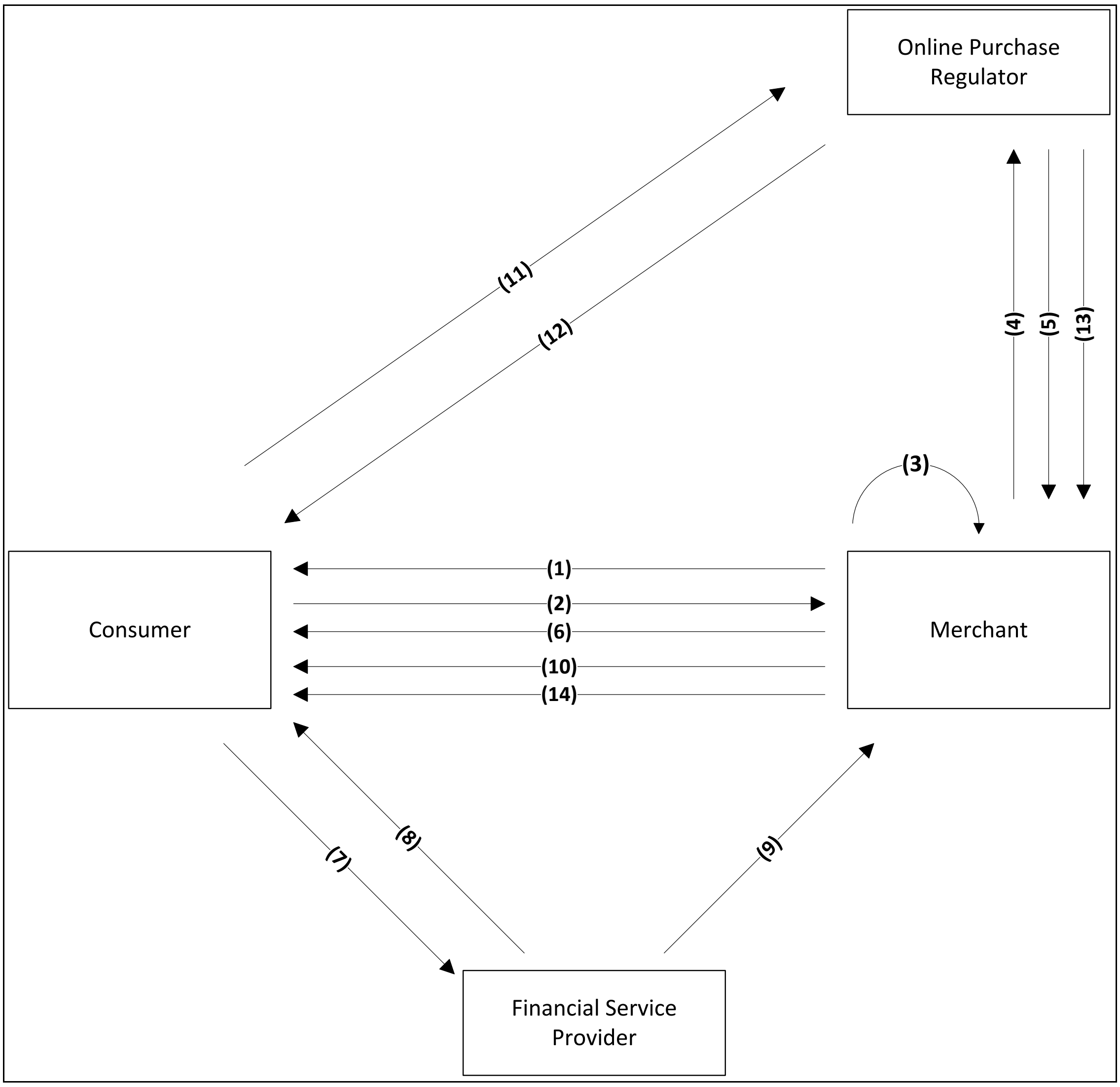
The merchant has to transfer the e-item, specifications and cost to a Trusted third party. The TTP then codes the e-item by utilizing the key KM_1 and then the product is promoted through advertisement on the Trusted third party's website. The buyer then goes to the Trusted third party's site and downloads the coded digital product. The communication between the transacting parties in the Ray et al protocol [26] comprises of four messages which have been summed up below.

The first electronic mail which is transmitted by the customer to the merchant consists of (1) the PO and (2) the coded P share the product key of KC_1 x KC_2. M then confirms the authenticity of the first message after getting it and if satisfied then M transmits to C the M2 with the e-item which is coded share the item key of KM_1 X KM_2. After receiving the second message, C makes a comparison of the coded digital product from the Trusted third party with the coded e-item which was received in the M2 buyer can be certain if the two compare well, then the un-coded e-item will be compared as well.

After making the comparison of the two e-item, if buyer is want keen on the exchange process then buyer transmits the M3 to seller with the decoding key for the coded payment. Lastly, the M4 is transmitted by seller to buyer with the decoding key of the coded e-item. In case of any solve problems, buyer should be communicate the Trusted third party.. According to this protocol, the Trusted third party. makes a duplicate for the e-item that seller intends to view. Hence the TTP has to create space for the storage of this extra data and also create extra security protection to safeguard the safety of the data. This creates a communication overhead.

Appendix C

This appendix includes the Promela code used in SPIN to formally verify the whole protocol include merchant model,customer model and FSP model.



```
#define TRUE 1
#define FALSE 0


/*
Protocol messages
*/
mtype = {M1,M2,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13,M14};


bool quitMerchant, quitConsumer;
bool paymentValidatedSuccessful, receiveCorrectProduct;


/* Merchant
Sends messages:M1, M4, M6, M10
Receives messages: M2, M5, M9
*/
proctype Merchant(chan channel_C_M, channel_M_FSP, channel_M_OPR)
{

    bool sendM1, sendM4, sendM6, sendM10, waitForM2, waitForM5, waitForM9;

    sendM1 = FALSE;

    sendM4 = FALSE;
    sendM6 = FALSE;
    sendM10 = FALSE;
    waitForM2 = FALSE;
    waitForM5 = FALSE;
    waitForM9 = FALSE;
    quitMerchant = TRUE;

    do

        :: (sendM1 == FALSE) ->
        channel_C_M!M1;
        sendM1 = TRUE;
        quitMerchant = FALSE;
        printf("Merchant: M1 sent to Consumer \n");

        if

            :: channel_C_M?M2 ->
            waitForM2 = TRUE;
            printf("Merchant: Recieves the M2 from the Consumer  \n");

            if
```

```
                :: TRUE -> /* Merchant decided the go-ahead with the
purchase  */
                quitMerchant = FALSE;
                channel_C_M!M4;
                sendM4 = TRUE;
                printf("Merchant: Merchant decided the go-ahead with
the purchase  \n");

                :: TRUE -> /* Merchant decided not to go-ahead with the
purchase  */
                quitMerchant = TRUE;
                printf("Merchant decided not to go-ahead with the
purchase  \n");
                break;

                :: timeout ->
                quitMerchant = TRUE;
                printf("Timeout at Merchant \n");
                break;

            fi

        fi

        :: (sendM1 == TRUE && sendM4 == TRUE) ->
        waitForM5 = TRUE;
        channel_M_FSP?M5;
        quitMerchant = FALSE;
        printf("Merchant: Received the confirmation from the regulator  \n");

        :: (waitForM5 == TRUE) ->
        sendM6 = TRUE;
        channel_C_M!M6;
        quitMerchant = FALSE;
        printf("Merchant: Sending the Invoice to the Consumer  \n");


        :: (sendM6 == TRUE && sendM10 == FALSE) ->

        if
            :: channel_M_FSP?M9 ->

            if
                :: (paymentValidatedSuccessful == TRUE) ->
                quitMerchant = FALSE;
                waitForM9 = TRUE;
                sendM10 = TRUE;
```

```
                                channel_C_M!M10;
                                printf("Merchant: Sending the product the consumer
\n");

                                :: (paymentValidatedSuccessful == FALSE) ->
                                quitMerchant = TRUE;
                                waitForM9 = TRUE;
                                channel_C_M!M10;
                                sendM10 = TRUE;
                                printf("Merchant: Quit since the payment is unseccesful.
Inform the consumer  \n");
                                break;

                                ::TRUE ->
                                quitMerchant = TRUE;
                                waitForM9 = TRUE;
                                channel_C_M!M10;
                                sendM10 = TRUE;
                                printf("Merchant: Quit since the payment is not done.
Inform the consumer  \n");
                                break;

                        fi

                fi



        od;

}



/* Consumer
Sends messages: M2, M7, M12
Receives messages: M1, M6, M8, M10, M14

*/
proctype Consumer(chan channel_C_M, channel_C_FSP, channel_C_OPR)
{

        bool sendM2, sendM7, sendM12, waitForM1, waitForM6, waitForM8,
waitForM10;

        sendM2 = FALSE;
        sendM7 = FALSE;
        sendM12 = FALSE;
```

```
    waitForM1 = FALSE;
    waitForM6 = FALSE;
    waitForM8 = FALSE;
    waitForM10 = FALSE;
    quitConsumer = TRUE;

    do

        :: (waitForM1 == FALSE) ->
        channel_C_M!M2;
        sendM2 = TRUE;
        waitForM1 = TRUE;
        quitConsumer = FALSE;
        printf("Consumer: M2 sent to Merchant \n");


        :: (quitMerchant == FALSE && waitForM1 == TRUE && waitForM6
== FALSE) ->
        channel_C_M!M6;
        sendM7 = TRUE;
        waitForM6 = TRUE;
        printf("Consumer: M7 sent to Finacial Service Provider \n");

        if

            :: TRUE -> /* Consumer like to go-ahead with the payment  */
            sendM7 = TRUE;
            quitConsumer = FALSE;
            printf("Consumer: Consumer like to go-ahead with the payment
\n");

            :: TRUE -> /* Consumer do not want to go-ahead with the
payment  */
            quitConsumer = TRUE;
            printf("Consumer: Consumer do not want to go-ahead with the
payment \n");
            break;

        fi

        :: (quitMerchant == FALSE && waitForM1 == TRUE && sendM7 ==
TRUE && waitForM8 == FALSE ) ->
        if
            :: (paymentValidatedSuccessful == TRUE) ->
            channel_C_FSP?M8;
            quitConsumer = FALSE;
            printf("Consumer: Gets the successful payment confirmation
\n");
```

```
                        :: (paymentValidatedSuccessful == FALSE) ->
                        channel_C_FSP?M8;
                        quitConsumer = TRUE;
                        printf("Consumer: Quit since the payment is unseccesful  \n");
                        break;


                fi

                :: (quitConsumer == FALSE && waitForM1 == TRUE && sendM7 ==
TRUE && waitForM6 == TRUE && waitForM8 == TRUE && waitForM10 ==
FALSE ) ->
                channel_C_M?M10;
                waitForM10 == TRUE;

                if

                        :: TRUE -> /* Consumer is not happy with the product  */
                        sendM12 = TRUE;
                        channel_C_OPR!M12;
                        receiveCorrectProduct = FALSE;
                        printf("Consumer: Is not happy with the product \n");

                        :: TRUE -> /* Consumer is happy with the product  */
                        receiveCorrectProduct = TRUE;
                        quitConsumer = TRUE;
                        printf("Consumer: Happy with the product \n");
                        break;

                fi

        od;

}


/* Financial Service Provider
Sends messages: M7
Receives messages: M8, M9


*/
proctype FinancialServiceProvider(chan channel_M_FSP, channel_C_FSP)
{


        do
                :: channel_C_FSP?M7 ->
                channel_C_FSP!M8;
                channel_M_FSP!M9;
                printf("Finacial Service Provider validates the payment details and send
the confirmation to Merchant and Consumer \n");

                if

                :: TRUE -> /* Payment is successfuly validated   */
                paymentValidatedSuccessful = TRUE;
                printf("Finacial Service Provider: Payment is successfuly validated");


                :: TRUE -> /* Payment couldn't be validated   */
                paymentValidatedSuccessful = FALSE;
                printf("Finacial Service Provider: Payment couldn't be validated");

                fi


        od;

}

/* Online Purchas eProvider
Sends messages: M13, M14
Receives messages: M12


*/
proctype OnlinePurchaseProvider(chan  channel_M_OPR, channel_C_OPR)
{


        do

                :: channel_C_OPR?M12 ->
                printf("Consumer contacts the Online Purchase Provider since merchant
has not sent the product \n");

                if

                :: TRUE -> /* If user details are validated with the message   */
                paymentValidatedSuccessful = TRUE;
                printf("Online Purchase Provider: Sending the product to the
Consumer");
                channel_M_OPR!M13;
```

```
                channel_C_OPR!M14;


                :: TRUE -> /* If the user details are not validated with the message   */
                printf("Online Purchase Provider: Invaild product/merchant
information");
                channel_C_OPR!M14;
                break;

                fi


        od;

}



init
{


chan channel_C_M = [0] of {mtype};
chan channel_C_FSP = [0] of {mtype};
chan channel_M_FSP = [0] of {mtype};
chan channel_C_OPR = [0] of {mtype};
chan channel_M_OPR = [0] of {mtype};

run Merchant(channel_C_M, channel_M_FSP, channel_M_OPR);
run Consumer(channel_C_M, channel_C_FSP, channel_C_OPR);
run FinancialServiceProvider(channel_M_FSP, channel_C_FSP);
run OnlinePurchaseProvider(channel_M_OPR, channel_C_OPR);


}
```