ON SECURE ELECTRONIC AUCTION PROCESS OF GOVERNMENT DOMESTIC DEBT SECURITIES IN TURKEY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ATİLLA BEKTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CRYPTOGRAPHY

AUGUST 2013

Approval of the thesis:

ON SECURE ELECTRONIC AUCTION PROCESS OF GOVERNMENT DOMESTIC DEBT SECURITIES IN TURKEY

submitted by **ATİLLA BEKTAŞ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** _____

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** _____

Prof. Dr. Ersan Akyıldız
Supervisor, **Mathematics, METU** _____

Dr. Mehmet Sabır Kiraz
Co-supervisor, **TÜBİTAK BİLGEM UEKAE, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics, METU _____

Prof. Dr. Ersan Akyıldız
Mathematics, METU _____

Assist. Prof. Dr. Ali Aydın Selçuk
Computer Engineering, Bilkent University _____

Prof. Dr. Ferruh Özbudak
Mathematics, METU _____

Dr. Muhiddin Uğuz
Mathematics, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : ATİLLA BEKTAŞ

Signature :

ABSTRACT

ON SECURE ELECTRONIC AUCTION PROCESS OF GOVERNMENT DOMESTIC DEBT SECURITIES IN TURKEY

Bektaş, Atilla

Ph.D., Department of Cryptography

Supervisor : Prof. Dr. Ersan Akyıldız

Co-Supervisor : Dr. Mehmet Sabır Kiraz

August 2013, 99 pages

Auctions, today, have become an important part of electronic commerce. With the gradually increasing importance of confidentiality and privacy in auction modeling, considering these two concepts, various designs have been proposed to ensure secure transmission especially in sealed-bid auctions. However, to the best of our knowledge, there has not been many approaches to the Treasury auctions. Looking at the current systems, many countries including Turkey perform Treasury auctions mostly manually. While it can be seen that there have been almost no problems in the processes and the procedures have been operated successfully so far, letting all the bids be transferred to the system in clear text and the operations being realized on clear text show that from a cryptographic point of view, confidentiality and privacy are not guaranteed and that therefore with the ongoing advances and developments in technology, this makes the system more vulnerable to such potential threats. On the other hand, since the knowledge of individual bids is of great value to the others who may use this knowledge to better their positions, it becomes crucial that the confidentiality of all submitted bids and privacy concerns of all the bidders should be satisfied. In a secure electronic auction system, from a cryptographic point of view, it is possible to determine the winner or the winners without revealing any private information. Within this scope, to accomplish this, in this thesis, we propose a new and efficient secure

electronic auction model for Government Domestic Debt Securities (government bonds and treasury bills), satisfying both confidentiality and privacy, based on secure multi-party computation, secret sharing and threshold homomorphic cryptosystem. To the best of our knowledge, this is the first study applied on issuing Government Domestic Debt Securities via electronic auction method.

Keywords: Electronic auction, government domestic debt securities, secure multi-party computation, threshold homomorphic cryptosystem, confidentiality, privacy

ÖZ

TÜRKİYE'DE DEVLET İÇ BORÇLANMA SENETLERİNİN GÜVENLİ ELEKTRONİK İHALE SÜRECİ HAKKINDA

Bektaş, Atilla
Doktora, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Ersan Akyıldız
Ortak Tez Yöneticisi : Dr. Mehmet Sabır Kiraz

Ağustos 2013, 99 sayfa

İhaleler, günümüzde elektronik ticaretin önemli bir parçası haline gelmiştir. Gizlilik ve mahremiyetin ihale modellemelerinde öneminin gitgide artmasıyla birlikte özellikle kapalı-zarf teklif usulü ihalelerde güvenli iletimi sağlamak için bu iki kavramın da göz önünde bulundurulduğu çeşitli tasarımlar sunulmuştur. Ancak, Hazine ihaleleri için bildiğimiz kadarıyla pek yaklaşım olmamıştır. Mevcut sistemlere bakıldığında, Türkiye'nin de içerisinde olduğu birçok ülkede Hazine ihaleleri çoğunlukla manuel olarak yürütülmektedir. Süreçlerde şu ana kadar hemen hemen hiçbir problemle karşılaşılmadığının ve süreçlerin başarılı bir şekilde işletildiğinin görülmesi ve bilinmesiyle beraber; tekliflerin sisteme açık metinler halinde iletilmesi ve yine açık olarak işlenmesi, kriptografik açıdan gizliliğin ve mahremiyetin garanti altında olmadığını ve dolayısıyla teknolojinin de ilerlemesiyle ve gelişmesiyle birlikte sistemin, olası tehditlere karşı ne yazık ki müsait olduğunu göstermektedir. Bununla birlikte, ihalelerde teklif verecek olan bir tarafın diğer bir tarafa ait teklifi bilmesi, kendi teklifini iyileştirebilmesi açısından da büyük önem arz etmesinden dolayı gönderilen her bir teklif için gizliliğin ve her bir teklif sahibi için mahremiyetin korunması azami derecede önemli olmaktadır. Kriptografik açıdan düşünüldüğünde güvenli bir elektronik ihale sisteminde, aslında kazanan veya kazananlar, dışarı herhangi bir özel bilgi ifşası olmadan belirlenebilir. Bu kapsamda; bunu gerçekleştirmek için bu çalışmada çok taraflı güvenli hesaplama, sır paylaşımı ve eşik homomorfik şifreleme sistemi tekniklerini esas alan, gizlilik ve mahremiyetin sağlandığı, devlet iç borçlanma

senetleri (hazine bonoları, devlet tahvilleri) için yeni ve etkin bir güvenli elektronik ihale modeli önerilmektedir. Bu çalışma, bildiğimiz kadarıyla özellikle devlet iç borçlanma senetlerinin elektronik ihale yöntemi ile satışı üzerinde uygulanan bir ilk çalışma olma niteliğini taşımaktadır.

Anahtar Kelimeler: Elektronik ihale, devlet iç borçlanma senetleri, çok taraflı güvenli hesaplama yöntemi, eşik homomorfik şifreleme sistemi, gizlilik, mahremiyet

Dedicated to my beloved mother, father and sister.

ACKNOWLEDGMENTS

I would like to thank my thesis supervisor, Prof. Dr. Ersan Akyıldız, and Assist. Prof. Dr. Ali Aydın Selçuk, for their patient guidance, enthusiastic encouragement and valuable advices during the development and preparation of this thesis.

I would like to express my great appreciation to my thesis co-supervisor, Dr. Mehmet Sabır Kiraz, for his significant advices, valuable guidance, termless support, invaluable motivation, encouragement and kindness throughout both in my research and in my life and for being a very good friend. His willingness to give his time and to share his experiences, despite his busy schedule, has brightened my path.

I thank the other members of my committee, Assoc. Prof. Dr. Ali Doğanaksoy, Prof. Dr. Ferruh Özbudak and Dr. Muhiddin Uğuz, for their unconditional support and for the knowledge gained in their lectures and discussions.

My special gratitude goes to Dr. Turgut Hanoymak, from Cryptography Department at METU, for his motivation, assistance, comments and support that provided me determination and power throughout this study.

I wish to extend my warmest thanks to Dr. Murat Ak, from Computer Engineering Department at Bilkent University, for his valuable comments and suggestions in designing our proposed protocol, and to Kamil Otal, from Mathematics Department at METU, for his comments and the precious help he gave me.

I would like to thank Ahmetcan Öztürk, my work friend, for his help in implementation trials for the protocols, for his kindness throughout both in my research and in my life and for being a very good friend.

My special thanks also goes to Sedef Baran Gürbüz, my work friend, for her help in revising the English of this thesis and her kindness. I thank my work friends Mustafa Ufku Kandemir, Dr. Yasemin Atasoy, Serhat Sağır, Metin Sezer, Mehmet Tekin Arpacık and Ahmet Alper Aycan for their support and encouragement in my graduate study; also my top managers Aşkın Alıcı and İbrahim Ömer Gönül for their support, patience and allowance me to study when I needed.

I also thank the institute staff especially Nejla Erdoğdu, Cevher Durmuş and Serkan Demiröz for their administrative help throughout my graduate study period.

Last but not least, I would like to thank my family for their endless support, hope, care, understanding and patience both in my studies and in my life. It is to them that I dedicate this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

" The most important point of all things is its beginning."

— Eflatun

Governments have to determine the expenses of social needs and have to meet these expenses with the revenues obtained. Therefore, they prepare annual budget estimates for the revenues and expenses. In the case of public expenditures not meeting with the revenues, budget deficit occurs. To finance that deficit, the governments resort to borrowing methods. In recent years, the ordinary revenues have become insufficient to meet the increasing expenditures due to the broadening in activities of the governments, and there has been an increased tendency to government borrowing [115].

Borrowing is a financial method that almost every country refers in different periods in order to fulfill public services. The main reason of government debt is high level of public deficit. Public deficits can be financed by

- *Central Bank resources*,
- *external debt*,
- *domestic debt*

or with their combination. One of the obstacles on realizing price stability, which is the main objective of Central Banks, is to grant advance and to extend credit to Treasury and to public establishments and institutions. Such credits, while unsolicited by the economic units, lead to monetary expansion. In other words, that causes coining unrequited money and therefore this raises the level of current inflation. Due to these connections, using *the Central Bank resources* for public finance is not preferred in terms of stability [112] and has been abolished in Turkey by laws because of its negative effect on inflation [114]. In 2011, with the amendment to the article 56 titled "Operations prohibited for the Bank" of the Law on the Central Bank of the Republic of Turkey, the Central Bank may not grant advance and extend credit to the Treasury and to public establishments and

institutions [114]. Thus, coining unrequited money is abolished. Furthermore, the Central Bank may not purchase debt instruments issued by the Treasury and public establishments and institutions in the primary market which bore the same result indirectly, i.e., this also results in coining unrequited money, and the aims for the level of market interest rates of monetary policy may deviate [28].

In 1990's, elections in Turkey at very frequent intervals caused the political instabilities. Also the wrong economic policies during these years increased the debt burden. Political and economic crisis in the country has brought about the problem of confidence to the government. Because of this non-confidence, Turkey was perceived as a risky country and so had difficulties in finding *external resources* except with some very high interest rates and thus the government with needing urgent finance showed tendency to the *domestic resources*, i.e., domestic borrowing [8]. In Turkey and also in many developing countries, domestic borrowing is widely-used. Being seen as an easy method compared to the external borrowing and to the taxation, the borrowing when needed from domestic individuals and institutions is much preferred [115].

Domestic debt refers to the money lent to the government as mandatory or voluntary from individuals, private institutions or public authorities for a specific maturity date and interest. Domestic borrowing processes related to the domestic debt in Turkey are based on a set of legal documents which are laws, bylaws, decisions of the *Turkish Treasury*[1], regulations and guidelines.

In Turkey, the public sector borrowing requirements increased rapidly in the last three decades resulting in overload of domestic debt stock. The main tool that the Turkish Treasury, the authorized body, uses to borrow in domestic markets is to hold regular ***auctions*** for *Government Domestic Debt Securities* (GDDSs) in order to reduce the debt stock. Being raised to the level of 80% of auctions in the cash domestic debt stock, the importance of GDDSs auctions (or Treasury auctions) has been enhanced [93]. Other tools such as TAP (comes from *tapping*) Sales, Direct Sales and Public Offerings are also used when needed. Moreover, the *Turkish Treasury* and the *Central Bank of Turkey* are the two main institutions involved in the auctions where the Central Bank acts as a fiscal agent of the Treasury, i.e., as an intermediary and principal paying agent for the Treasury. Technical operations related with the auction process are carried out by the Central Bank on a computer network between the bidders and the Central Bank.

1.1 Scope of This Thesis

As we mentioned above, the Turkish Treasury holds regular ***auctions*** for GDDSs in order to borrow in domestic markets. In the current practice, the most important bidders in Turkey are the *banks*. The lack of credit risk, relatively high

[1] *Turkish Treasury* will refer to *Republic of Turkey Prime Ministry Undersecretariat of Treasury* throughout this thesis.

interest rates, being much used on the secondary markets, usage as a collateral by the auctions, no risk of non-payment, and being as a subject of open market operations make GDDSs more preferable by the banks. The banks which are authorized in auction process are called *Primary Dealers*. There are currently 13 primary dealers in Turkey for 2013 period[2] and the latest list of these banks are announced on the Treasury website[3] if there is an update on the list. Throughout this thesis, without loss of generality, only those banks, i.e., primary dealers, are considered as the bidders on auctions.

In these auctions, ***unencrypted*** bids are submitted by the bidders to the Central Bank by means of ***conventional ways***, e.g., EFT[4], TETS[5] and fax. In fact, the banks use EFT channel while the other bidders which are minority use the other two channels. After that bidding step, the valid submitted bids are sorted from higher price offered to lower price and then the new ordered list is transmitted to the Treasury. The Treasury then examines and evaluates the submitted quotes, and finally determines the winners. The determination process is done ***manually*** by the Treasury experts.

While it is seen and known that there have been almost no problems in the auction processes and the procedures have been operated successfully so far, letting all the bids be transferred in ***clear text*** and the operations being realized on clear text show cryptographically that ***confidentiality*** of all the submitted bids and ***privacy*** concerns of all the bidders are not guaranteed and with the advances and developments in technology, this makes the system vulnerable to potential threats. For example, a *corrupted user on the Central Bank* may share some of the bids with other parties or bidders since he/she can see all the submitted bids. Similarly, a *corrupted user on the Treasury* may change the order of the accepted/rejected bidders in the list, i.e., may replace the final result with another loser with no detection. Another example, if *fax channel is corrupted* it also causes to be a security violation as all the transferred bids are in clear text. As it is seen, the manual system is insecure from the cryptographic point of view. Moreover, since the knowledge of individual bids is of great value to the others who may use this knowledge to better their positions it becomes crucial that the confidentiality of all the submitted bids and privacy concerns of all the bidders should be satisfied.

In this thesis, we mainly focus on ***improving the current manual auction system*** by ***proposing a secure electronic system*** where all the bids (offered prices and offered amounts) and the corresponding name of the bidders are kept secret until the auction result is published. Confidentiality of the bids are assured under the assumption that the Treasury and the Central Bank do not cooperate. While the system is easily usable in other similar scenarios, we examine the whole process from bid submission to auction award securely using the underlying cryptographic techniques which are *secure multi-party computation (MPC)*, *secret*

[2] Press Release dated December 18, 2012 and No.2012/199 on *Primary Dealers for 2013 Period*.
[3] Turkish Treasury, http://www.treasury.gov.tr
[4] EFT: Electronic Funds Transfer
[5] TETS: Takasbank Electronic Transfer System

sharing and *threshold homomorphic cryptosystem*. We also use *secure sorting with secure comparison* as a subprotocol whose algorithms can be found in Chapter 4 and Appendix A.

1.2 Contributions

Current auction process of Turkey is performed manually. The bidders submit their bids in clear text to the Central Bank. After the auction deadline, all the bids are sorted by price and the created ordered list is sent again in clear text to the Treasury. The Treasury, then, reviews and evaluates the ordered bids and determines the winners. Despite there have been almost no problems in processes and the procedures have been operated successfully so far, from the cryptographic point of view, confidentiality and privacy are not guaranteed in the current system. This means, it is possible to manipulate the results in the case of curious adversaries.

In this thesis, we propose a *new solution* for GDDSs of Turkey by using *secure MPC*, *secret sharing* and *threshold homomorphic cryptosystem*. We initially describe the current manual system and the drawbacks, and then outline our proposed model and finally explain the reasons why secure MPC turned out to be a good solution.

Our proposed protocol includes the following three parties:

- *Primary Dealer*,
- *Central Bank*,
- *Treasury*.

These three parties interacts with each other (see Figure 1.1) via two phases which are explained in details in Chapter 4:

- Submission and Evaluation phase,
- Award phase.

The Submission and Evaluation phase obtains the encrypted bids from authentic bidders and performs secure sorting with secure computation on those values on the side of the Central Bank. After those operations, the Treasury is given all the bids which pass the acceptance criteria. In this phase, confidentiality of bids and privacy of bidders are protected by cryptographic techniques *under the assumption that Treasury and Central Bank do not collude*. On the other hand, the Award phase clarifies the accepted primary dealers and publishes the statistical calculations. During the Award phase, each bidder can learn only its own result whereas the Treasury can learn all the results without knowing

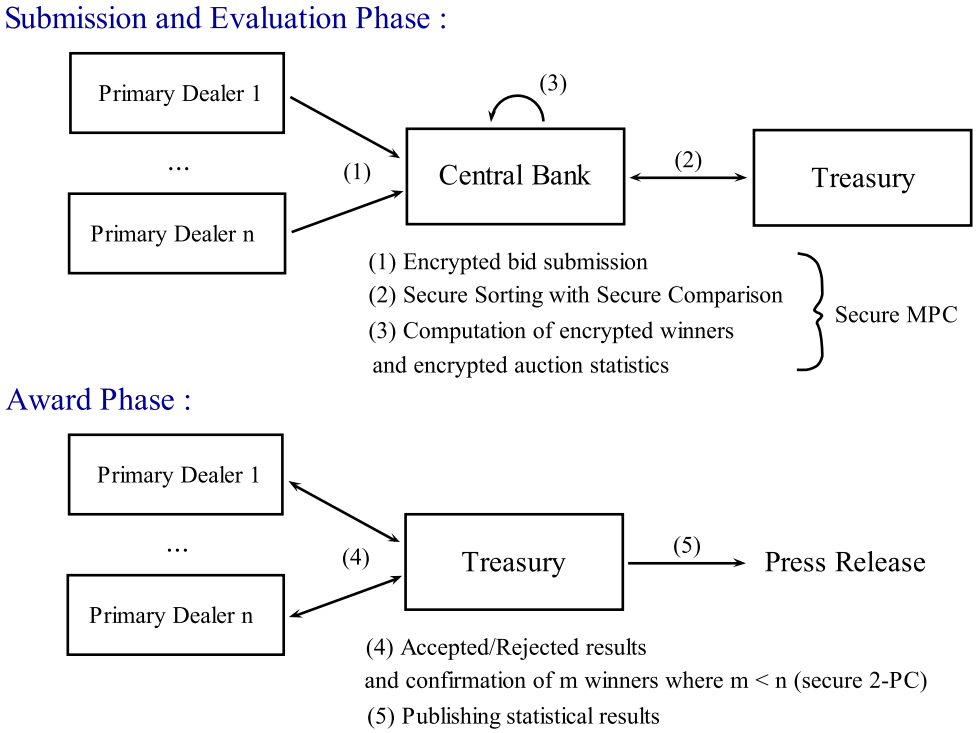

Figure 1.1: Interaction of the Participants in a Treasury Auction

the corresponding primary dealer's identity except of the winners. Finally, the winners send their confirmations to the Treasury to check that the bids and the corresponding bidders are matched correctly.

To the best of our knowledge, this is the first study applied on issuing domestic debt securities via electronic auction method in which a secure electronic auction process is included and both confidentiality and privacy are satisfied.

1.3 Thesis Outline

This thesis contains five chapters. The knowledge is constructed in the sequence of reviewing the current state, understanding the mechanisms, defining the problems of the current system and designing the new and secure protocol.

In *Chapter 1*, we give an introduction to the subject and give a summary to the readers. The details are outlined in the following chapters.

Chapter 2 is a review of financial and cryptographic preliminaries. It supports the understanding of building blocks for constructing our proposed protocol. This chapter reviews a set of related financial concepts such as government debt, domestic borrowing, domestic debt securities, their sales methods and auction; and related cryptographic technologies including public key cryptosystem with some examples, cryptographic hash functions, threshold cryptography, homomorphic

encryption, digital signature and secure multi-party computation.

Chapter 3 presents a detailed information about auctions and presents some selected cryptographic auction protocols with comparing them for domestic borrowing case. Moreover, some information on Treasury auctions of US, UK, Germany and Turkey are mentioned. The Turkish Treasury auctions in details including some definitions and some rules in offering mechanism are also outlined in this chapter. The Turkish Treasury auction mechanism is presented with an example for ease of understanding.

Chapter 4 introduces our proposed model in details, and security analysis and complexity analysis are presented here.

Finally, *Chapter 5* draws conclusions, summarizes the study and discusses the generalizations of our proposed model.

CHAPTER 2

PRELIMINARIES

" I am always doing that which I cannot do, in order that I may learn how to do it."

— Pablo Picasso

In this chapter, we present the definitions and concepts that are used in this thesis. We also review some financial and some cryptographic primitives needed to develop our proposed model. Additional illustrations and notations of some specific parts of the thesis occurs in the following chapters.

2.1 Financial Preliminaries

2.1.1 Government Debt

Government debt (also named as public debt or national debt), which was a subject of scientific investigation for the first time in 18th century, has gradually become an attractive issue in public finance. Debt and debt management has a significant share in modern public finance. Today, government debt is of great importance from the point of developing, developed and least developed countries [115].

When making state budget estimates in public economy, first thing is to make expenditure estimates, after that, means of revenue generation to meet these expenditures are sought. In case revenues and expenditures are equal, it can be said there is a balanced budget. In the case of differences between revenues and expenditures, there is budget deficit or budget surplus [34]. Budget deficits of public are financed by three basic ways in addition to supplementary levies [112]. These are;

- *utilization of central bank resources*,
- *foreign debt*,

- *domestic borrowing.*

Utilization of central bank resources means coining money. As mentioned in Chapter 1, using this method was abolished by law in 2001 [114]. *Foreign debt* is to obtain financing from international institutions and foreign markets usually in order to finance balance of payment and implementation of certain projects. That foreign debt possibilities have decreased gradually since the early 1990's have led *domestic borrowing* a more commonly used tool [115].

2.1.2 Domestic Borrowing

Domestic borrowing is a process of obtaining resource on voluntary basis from domestic markets at certain maturity and interests. Domestic borrowing is values which are taken from lenders on their own will by paying certain sums and to be repaid to them at the end of the maturity period [115]. When looked at the evolution of domestic borrowing in Turkey, it can be seen that domestic borrowing which was used especially for financing the development since 1930's was no longer popular until the end of 1970's. It has been an important financial resource that has been commonly used for financing public deficits since the early years of 1980's [113].

The most important change in borrowing policies in Turkey was made in 1986. With separation of Treasury from the Ministry of Finance, debt management is tried to be an independent administrative body [47]. Another change in policy since 1986 is the use of more borrowing especially domestic borrowing in order to close growing financial deficits. Treasury began to hold auctions for selling bonds and securities in 1986. Selling of government bonds through auctions has made it possible to cover most of the public deficits from money and capital markets, therefore the need to resort to central bank resources has decreased significantly [115].

The domestic borrowing process in Turkey is based on a set of legal documents which are laws, bylaws, decisions of the Turkish Treasury, regulations and guidelines. The most important of these documents is the Law on Regulating Public Finance and Debt Management, No. 4749 issued on March 28, 2002 and amended on July 31, 2003. In this law, *Government Domestic Debt* is defined as "Domestic Debt Securities issued by the Treasury within the country, Treasury's borrowings from domestic market in order to meet its temporary cash requirement, and all kinds of financial obligations assumed by the Treasury, regardless of whether the same are based on a note".

2.1.3 Domestic Debt Securities

Domestic Debt Securities expresses government securities issued by the Treasury domestically. They are sold through the Central Bank. The issued *government*

domestic debt securities (GDDSs) can be classified into five categories according to maturities, the type of currency in issuance, whether having coupon on them, the type of interest payment and issuance methods [82].

1. It can be classified in two ways as Government Bond and Treasury Bill depending on the maturity.

   - *Government Bonds:* Government Bonds are debt securities whose maturities are one year (364 days) or more as from the date of their issuance [121]. These are domestic borrowing bonds with interest coupons that have reimbursement once in 3 or 6 months and minimum maturity of one year or more as from first issuance [112]. Selling is made at prices determined by the Central Bank on basis of interest coupon on the bond and desired return [115]. They can be classified in various forms according to issuance specifications.
   - *Treasury Bills:* Treasury Bills are debt securities whose maturities are less than one year (up to 364 days) as of the date of their issuance [121]. Treasury Bonds does not include periodical interest payments but they can be bought and sold below their nominal values. Value of interest the investor has gained constitutes the difference between nominal price and the price of the treasury bill at maturity date [60]. Treasury bills issued since 1985 according to discount base and through auction are with maturity of 3, 6, 9 months and the interest rates are determined during the auction [115].

2. Issuance of Domestic Government Bond can be made in *Turkish Lira* or *foreign currencies*. The bonds issued in foreign currencies can take its part in domestic debt stock as foreign currency or foreign currency indexed. The most significant difference between currency and currency indexed bonds is that transactions regarding borrowing and payments of bonds in foreign currency are made in that currency while transactions regarding borrowing and payment of currency indexed bonds are made in Turkish Lira which is calculated according to the determined exchange rate [82].

3. Classification according to coupon situation can be made as *coupon* or *zero coupon* bonds. Zero-coupon bonds (also called discount bond) can be described as the "fixed yield bonds" that are sold at discount price at the issuance date and that have no coupon payment during its maturity period and whose return for the investor is the difference between nominal price and discounted price. Coupon bonds are the bonds that provide cash flow to its investors who keep the bonds until their redemption date. Coupon bonds can be issued at nominal prices as well as "discounted" below nominal prices and "incremental" over the nominal price [82].

4. Classification according to the type of interest can be made as *fixed* and *variable*. Discounted bonds are fixed yield bonds. Coupon bonds are issued with fixed or variable interest payments. Fixed interest bonds guarantee a certain income provided that the investor holds the bond until the maturity

date and the income that the investor will gain is known at the beginning of maturity. Variable interest payment bonds enable to reduce the risks assumed by the investors associated with excessive interest rate volatilities that might be occurred between issuance and redemption date and they also enable long run loan for borrowers.

5. According to issuance methods, four basic methods are used. These are;

   - *auction*,
   - *float (TAP)*,
   - *direct sale*,
   - *public offer*.

   In *auction method*, borrowers give their written proposals regarding unit amount and requested price to the Central Bank until 12 o'clock on the day of the auction. The Treasury concludes the auction after examining the proposals. In *float method (TAP)*, long dated variable interest bonds issued by the Treasury are sold by the Central Bank. In *direct sale method*, the amount of bond and the buyer is determined at the beginning. With the issuance of the bond, selling process is completed [93]. In *public offer method*, selling conditions are determined by the Treasury and they can be sold through banks and the Central Bank. In case other methods different from the auction method are needed, this method can be applied.

2.1.4 Auction

An auction is a method or mechanism of selling through bidding in a public competition. Indeed it is a *game* with partial information where a party's appraised value of an object is kept secret from other parties. According to McAfee and McMillan [99], an auction is "*a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants.*". In a more general expression, an *auction* is a market mechanism with certain rules which can be held separately or simultaneously for the purposes of selling or buying, and determining and showing the prices of items in the market that have no standard value accepted by everyone [93].

When looked at auction samples from the view of proposal structure; in *auctions held by sellers*, buyers give their proposals in order to get the chance to buy at minimum price while in *auctions held by buyers*, sellers shape their proposals to obtain the highest price level. However, in practice there are auctions which can be regarded as *double sided auctions* in which sellers give their proposals to sell the items and buyers give their proposals to buy the same items. In double sided auctions, buyers declares their proposal of price that they would like to pay for the items and sellers states the price that they would like sell their items. The intersection between demand function created from buyers' proposals and supply function created from the proposals of sellers will enable equilibrium price and amount level to be formed [58].

In next chapter, we present the types of auction and some known auction schemes with some selected Treasury auctions in order to understand the "auction concept" in more details.

2.2 Cryptographic Preliminaries

In modern terms, indirect communication makes it harder to ensure *confidentiality*, *authenticity* or *integrity* of messages. These concerns very naturally lead to the development of techniques diminishing them. Traditionally, problems with authenticity and integrity were usually solved by trusted messengers, while the confidentiality issue was addressed by algorithmic and/or technological means. This is where the name *cryptography* stems from: the practice of secret writing. The contemporary use of the term cryptography also includes techniques for authenticity and integrity of communications. Modern cryptography captures the objective of achieving confidentiality in the notion of *cryptosystems* [138].

Definition 2.1. A *cryptosystem* is a tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ satisfying the following properties:

1. $\mathcal{P}$ is a finite set of plaintexts,
2. $\mathcal{C}$ is a finite set of ciphertexts,
3. $\mathcal{K}$ is a finite set of keys,
4. $\mathcal{E} = \{\mathcal{E}_k : k \in \mathcal{K}\}$ is a family of encryption functions $\mathcal{E}_k : \mathcal{P} \to \mathcal{C}$,
5. $\mathcal{D} = \{\mathcal{D}_k : k \in \mathcal{K}\}$ is a family of decryption functions $\mathcal{D}_k : \mathcal{C} \to \mathcal{P}$,
6. For any key $e \in \mathcal{K}$ there exists another key $d \in \mathcal{K}$ such that we have $\mathcal{D}_d(\mathcal{E}_e(p)) = p$ for all plaintexts $p \in \mathcal{P}$.

The following definition given by Schoenmakers [127] shows the distinction made between

- *cryptographic algorithms*,
- *cryptographic protocols*, and
- *cryptographic schemes*.

Definition 2.2. A *cryptographic algorithm* is a well-defined relation where an output value is produced with a given input value accomplishing predetermined security aims. A *cryptographic protocol* is a distributed algorithm having interactions between two or more parties accomplishing predetermined security aims. A *cryptographic scheme* (or a cryptographic system) is a set of related cryptographic algorithms and cryptographic protocols accomplishing predetermined security aims.

Parties (or entities) act in a cryptographic protocol by sending and receiving data or messages between each other over special communication channels [127] such as

- *point-to-point channels*, and
- *broadcast channels*.

Point-to-point channels connect two sides, whereas *broadcast channels* connect one sender to multiple receivers. Communication channels are usually represented by the security guarantees [127] as

- *private channel* (or, *secure channel*),
- *public channel*,
- *bulletin board*.

A *private channel* is a point-to-point channel using encryption and authentication techniques to secure the channel and to secure the exchanged messages against eavesdropping. A *public channel*, on the other hand, is a channel lacking the protection techniques as in the private channel, i.e., no protection against eavesdropping. A *bulletin board* is a public authenticated broadcast channel. This is a type of channel in which a sender is allowed to broadcast an authenticated message [127].

A party or an entity which is included in a process of a cryptographic protocol may be

- *honest*, or
- *dishonest*.

A *honest* party follows the protocol every time exactly as described and does nothing else. On the other hand, a *dishonest* party is supposed to be managed by an adversary, corrupting a party and getting all the information known to that party [89]. Adversaries are coalitions composed of an attacker and/or one or more of the parties involved in the cryptographic scheme who can be either

- *passive* (also known as *semi-honest* or *eavesdropper*), or
- *active* (also known as *malicious*).

A *passive* adversary follows the protocol specifications but tries to extract some extra information by examining the messages that is sent and received during the protocol execution. On the other hand, an *active* adversary takes all the control

over corrupted parties those who act arbitrarily and do not follow the protocol specifications according to the instructions of the adversary [89].

An adversary may also be

- *static*, or
- *adaptive* (also known as *dynamic*).

A *static* adversary determines the parties who will be corrupted before the protocol starts, and those parties remain unchanged during the protocol execution. An *adaptive* adversary, in other respects, can select the parties dynamically during the protocol execution, i.e., at any time during the computation. This is sometimes dependent on the information he/she takes before determining which party he/she is going to corrupt [89].

Furthermore, there are two traditional security classes.

- *computational security*, or
- *unconditional security* (also known as *information-theoretic security*).

Informally, *computational security* of a system is a case when the adversary's effort to break the system is computationally infeasible [62], i.e., the adversary is computationally limited or it is a polynomial-time adversary [132]. On the other hand, *unconditional security* or *information-theoretic security* of a system is a case when the system cannot be broken given infinite computational resources. Consequently, unconditional security is stronger than computational security [62].

After discussing the above security related concepts, we can further present other concepts such as symmetric and asymmetric cryptosystems with cryptographic hash functions and digital signatures which are among the most popular and standardized cryptographic mechanisms. They have wide application in for example electronic auction protocol design [53]. To review these mechanisms in advance is necessary and can reduce the repetition of discussion in following chapters. Secure MPC and threshold homomorphic cryptosystem which are discussed later in this chapter are more specific to electronic auction applications and their relation to secure electronic auction protocol design will be discussed in next chapters.

2.2.1 Symmetric Key Cryptosystem

Symmetric encryption, also known as conventional encryption or single key encryption was the only type of encryption in use before 1976 when public key encryption is developed. As shown in the Figure 2.1, *symmetric key cryptosystem* has five components, $(p, c, k, \mathcal{E}_k, \mathcal{D}_k)$.

- p (plaintext), chosen from the plaintext space $\mathcal{P}$, is the original intelligible message or data that is used in the algorithm as input.

- c (ciphertext), chosen from the ciphertext space $\mathcal{C}$, is the scrambled and unintelligible output message. The plaintext and the key are needed to form this output, i.e., $\mathcal{E}_k(p) = c$ with $k \in \mathcal{K}$, $p \in \mathcal{P}$ and $c \in \mathcal{C}$.

- k (secret key), chosen from the key space $\mathcal{K}$, is an input of the encryption algorithm and must be kept secret by all participants involved in the system.

- $\mathcal{E}_k$ (encryption algorithm) implements various operations on the plaintext p with the secret key $k \in \mathcal{K}$.

- $\mathcal{D}_k$ (decryption algorithm) is mainly the encryption algorithm run in reverse. It uses the ciphertext $c \in \mathcal{C}$ and the secret key $k \in \mathcal{K}$ as an input and outputs the original plaintext $p \in \mathcal{P}$, i.e., $\mathcal{D}_k(c) = p$.

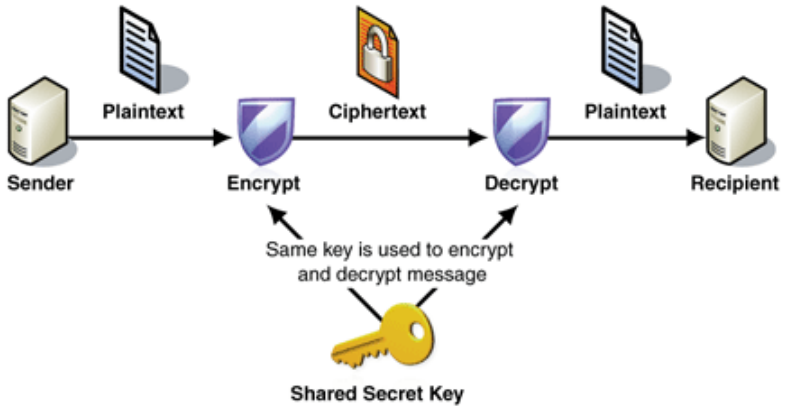

Figure 2.1: Symmetric Encryption Process [102]

As illustrated in Figure 2.1, symmetric encryption involves two participants: *Sender* and *Recipient*. With symmetric cryptography, both the sender and the recipient share a key which is used in both encryption and decryption. It is not needed to keep the encryption and decryption algorithms secret but it is strictly a must to keep the key secret. Otherwise if someone has this key and knows the algorithm, he/she can read all communications using this key [134].

Rijndael (AES) [42], Triple DES (3DES), IDEA and RC4/RC5 are the most well known symmetric encryption algorithms.

Symmetric cryptography is used to share information between a set of people that all will have access to it. Furthermore, it is easy to understand (less complex) and the algorithms tend to be faster. However, the sender and the receiver must agree on a secret key before the communication starts. Sometimes, for example in SSL, asymmetric cryptography is used to exchange the initial key over a secure channel.

2.2.2 Asymmetric Key Cryptosystem

In symmetric key cryptosystem, distribution of the secret key which is going to be shared by the sender and the recipient is a problem. If there exists n people communicating with each other, $\frac{n(n-1)}{2}$ symmetric keys between them are needed to distribute. The problem is to find a way to decrease the number of required shared keys.

In 1976, bringing in something new, Diffie and Hellman from Stanford University developed a method that addressed the above problem and that was ineradicably different from all previous approaches to cryptography [134]. We can describe the components of a public-key cryptosystem with a six-tuple array $(p, c, pk, sk, \mathcal{E}_{pk}, \mathcal{D}_{sk})$ as seen in Figure 2.2.

- p (plaintext), chosen from the plaintext space $\mathcal{P}$, is the original intelligible message or data that is used in the algorithm as input.
- c (ciphertext), chosen from the ciphertext space $\mathcal{C}$, is the scrambled and unintelligible output message. The plaintext and the recipient's public key are needed to form this output, i.e., $\mathcal{E}_{pk}(p) = c$ with $pk \in \mathcal{K}$, $p \in \mathcal{P}$ and $c \in \mathcal{C}$.
- (pk, sk) (public key, private key) is a pair of keys, both chosen from the key space $\mathcal{K}$. They are selected so that if one of the keys is used for encryption, the other key is used for decryption. Sometimes, the expression "secret key" is used instead of "private key". In order to avoid confusion, in general, the abbreviation sk is used standing for "private key".
- $\mathcal{E}_{pk}$ (encryption algorithm) implements various operations on the plaintext p with the recipient's public key $pk \in \mathcal{K}$.
- $\mathcal{D}_{sk}$ (decryption algorithm) uses the ciphertext and the recipient's secret key $sk \in \mathcal{K}$ to produce the original plaintext $p \in \mathcal{P}$, i.e., $\mathcal{D}_{sk}(c) = p$.

As illustrated in Figure 2.2, with *asymmetric key cryptography* which is also known as public key cryptography, the sender encrypts the given plaintext with one key to produce the ciphertext, and the recipient uses the other key to decrypt that ciphertext. The keys used in encryption and decryption are often referred to as a *public/private key pair*.

The basic steps followed by the sender and the recipient in a public key cryptosystem are the following [134]:

1. The sender generates a pair of keys, say (pk_1, sk_1), to be used for encryption and decryption of messages. We assume that the recipient also does so independently, i.e., generates (pk_2, sk_2).
2. The sender places his/her public key pk_1 in a public register or in a key distribution center, or his/her own personal web page. The companion key sk_1 is kept private. The recipient does so too.

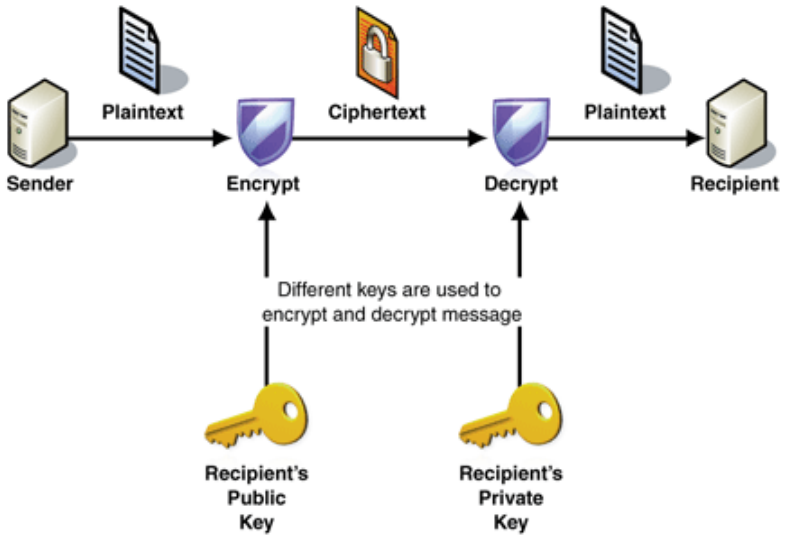

Figure 2.2: Asymmetric Encryption Process [102]

3. If the sender wants to send a plaintext (i.e., a confidential message), say p, to the recipient, the sender encrypts it by the encryption algorithm using the recipient's public key, i.e., $\mathcal{E}_{pk_2}(p) = c$, yielding corresponding ciphertext.

4. When the recipient receives the ciphertext, he/she decrypts it using his/her private key (to which only he/she has access), i.e., $\mathcal{D}_{sk_2}(c) = p$.

As seen from the basic steps, all participants can reach the public keys, and private keys need never be distributed. Each participant communicates with each other by using four keys (two private/two public). In addition, each participant utilizes the same public/private key pairs while communicating with other participants, i.e., one no longer needs $\frac{n(n-1)}{2}$ keys to be exchanged [134].

RSA [123] is the most commonly used asymmetric algorithm[1]. ElGamal [57], Paillier [116] (which we use in this thesis) and Cramer-Shoup [38] are some other examples of asymmetric algorithms. Also various protocols use asymmetric key algorithms such as PGP[2], SSH[3], SSL[4], ZRTP (a secure VoIP protocol)[5].

---

[1] Sometimes plain RSA or textbook RSA is used instead of RSA.

[2] *Pretty Good Privacy* (PGP) provides confidentiality and integrity in e-mail communications utilizing digital signatures, encryption and compression techniques [27]. PGP supports some algorithms such as IDEA, RSA, DSA, MD5 and SHA-1.

[3] *Secure Shell* (SSH) is a cryptographic network protocol, via a secure channel over an insecure network, for secure data communication, remote command-line login, remote execution of commands, and other secure network services between two networked computers that connects a server and a client [149].

[4] *Secure Sockets Layer* (SSL) is a security technology that establishes an encrypted link between server and a client.

[5] ZRTP (composed of *Z and Real-time Transport Protocol*) is a cryptographic key-agreement protocol [153].

2.2.2.1 RSA Cryptosystem

Being a public key encryption scheme, RSA [123] uses the modulo function which is a one-way function. It is invented by Ron Rivest, Adi Shamir and Leonard Adleman[6] in 1977. The scheme (Figure 2.3) works as follows.

Public key	:	(n, e)
Private key	:	(n, d)
Encryption	:	plaintext $m < n$ ciphertext $c = m^e \mod n$
Decryption	:	ciphertext $c < n$ plaintext $m = c^d \mod n$

Figure 2.3: RSA Encryption Scheme [123]

Key Generation:

The RSA encryption process uses two keys, public key and private key. The public key is used for encrypting plaintexts in order to generate ciphertexts. Ciphertexts are then decrypted using the corresponding private key. The steps to obtain public and private keys are outlined in the following.

1. Let p and q be two large random prime numbers.
2. Calculate $n = pq$.
3. Calculate $\varphi(n) = (p-1)(q-1)$.
4. Choose $e \in \mathbb{Z}$ such that:
   (a) $1 \leq e \leq \varphi(n)$
   (b) $gcd(e, \varphi(n)) = 1$, i.e., e and $\varphi(n)$ are co-prime.
5. Find d such that $ed \equiv 1 \mod \varphi(n)$.

As a result, the public key is (n, e) and the private key is (n, d).

Encryption:

In RSA encryption method, plaintexts are converted into sequences of integers. This conversion can be done by translating each letter into an integer, as is done with the Caesar cipher[7]. These integers are combined together to form

[6] *RSA* stands for Rivest, Shamir, and Adleman, the names of its inventors.

[7] *Caesar cipher* is a substitution cipher replacing each letter of the plaintext with a different letter which is a fixed number of positions further in the alphabet.

larger integers, each representing a block of letters, i.e., segments, say s_i's with $1 \leq s_i \leq n-1$ and $i \in \mathbb{Z}^+$. Formally, using American Standard Code for Information Interchange (ASCII) encoding of the upper case English letters, we get the following encoding table. For the whole table see Appendix B.

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

Table 2.1: ASCII Encoding of Upper Case English Letters

To find the encryption of the entire plaintext m, the sender first encrypts all those s_i's as

$$\mathcal{E}_{pk}(s_i) = s_i^e \mod n$$

by using the recipient's public key pk and then concatenates all of them.

Decryption:

To recover the plaintext m, the recipient uses blocks of numbers in ciphertext. Using his/her private key sk, the recipient raises each block to the power of d, reduces the result modulo n as

$$\mathcal{D}_{sk}(m_i) = m_i^d \mod n$$

and then concatenates all of them.

Example 2.1. Suppose a sender wants to encrypt the plaintext `HELLO` using RSA encryption scheme. From the above ASCII table (Table 2.1), the plaintext message `HELLO` is mapped to integers of the ASCII encodings as given below. Thus, concatenating all the integers, the given plaintext can be represented by the integer $m = 7269767679$.

H	E	L	L	O
72	69	76	76	79

Setup:

1. Let $p = 11$ and $q = 17$ be two prime numbers. In this example, we choose small primes for simplicity. Whenever a secure code is aimed, the larger the primes the better.
2. $n = pq = 11.17 = 187$.
3. $\varphi(n) = \varphi(187) = (11-1)(17-1) = (10)(16) = 160$. $\varphi(n)$ must not be published if the code is to remain secure.
4. Let $e = 7$ satisfying the restrictions

(a) $1 \leq e \leq \varphi(n)$

(b) e and $\varphi(n)$ are co-prime.

5. Then $d = 23$ satisfying $ed \equiv 1 \mod \varphi(n)$ [8].

As a result, the public key is $(n, e) = (187, 7)$ and the private key is $(n, d) = (187, 23)$. Note that, these keys are the recipient's keys.

Encryption:

1. The sender divides the numeric plaintext into segments. Each segment must contain the largest possible number less than $n = 187$. That is,

$$s_1 = 72, s_2 = 69, s_3 = 76, s_4 = 76, s_5 = 79.$$

Then, using the recipient's public key $pk = (n, e) = (187, 7)$, the sender gets

$$\begin{aligned}
\mathcal{E}_{pk}(s_1) = \mathcal{E}_{pk}(72) = 72^7 \mod 187 = 30 \\
\mathcal{E}_{pk}(s_2) = \mathcal{E}_{pk}(69) = 69^7 \mod 187 = 86 \\
\mathcal{E}_{pk}(s_3) = \mathcal{E}_{pk}(76) = 76^7 \mod 187 = 32 \\
\mathcal{E}_{pk}(s_4) = \mathcal{E}_{pk}(76) = 76^7 \mod 187 = 32 \\
\mathcal{E}_{pk}(s_5) = \mathcal{E}_{pk}(79) = 79^7 \mod 187 = 139.
\end{aligned}$$

2. The sender concatenates all the outputs and gets the ciphertext as

$$\mathcal{E}_{pk}(m) = 30863232139$$

and sends this value to the recipient.

Decryption:

1. The recipient takes the ciphertext $c = 30863232139$ which can be divided into segments each of which is less than $n = 187$ as

$$m_1 = 30, m_2 = 86, m_3 = 32, m_4 = 32, m_5 = 139.$$

[8] $ed \equiv 1 \mod \varphi(n)$ and $1 \leq e \leq \varphi(n)$ implies that $ed - 1 = k.\varphi(n)$. The value of $e = 7$ is publicized. However, only the recipient knows the value of $\varphi(n) = 160$. Therefore, he/she can plug these values into the equation $ed - 1 = k.\varphi(n)$ and obtain $7d - 1 = 160k$. Besides, taking $k = 1$, he/she gets $160 = 7.22 + 6$
$\Leftrightarrow 6 = 160 - 7.22$
$\Leftrightarrow 1 = 7 - (160 - 7.22)$
$\Leftrightarrow 1 = 7.23 - 160$
$\Leftrightarrow 7.23 - 1 = 160$
$\Leftrightarrow 7.23 = 1 \mod 160$, which yields $d = 23$.

Then, the recipient uses his/her private key $sk = (n, d) = (187, 23)$ and gets

$$\begin{aligned}
\mathcal{D}_{sk}(m_1) &= \mathcal{D}_{sk}(30) = 30^{23} \mod 187 = 72 \\
\mathcal{D}_{sk}(m_2) &= \mathcal{D}_{sk}(86) = 86^{23} \mod 187 = 69 \\
\mathcal{D}_{sk}(m_3) &= \mathcal{D}_{sk}(32) = 32^{23} \mod 187 = 76 \\
\mathcal{D}_{sk}(m_4) &= \mathcal{D}_{sk}(32) = 32^{23} \mod 187 = 76 \\
\mathcal{D}_{sk}(m_5) &= \mathcal{D}_{sk}(139) = 139^{23} \mod 187 = 79
\end{aligned}$$

2. Concatenating all the outputs, decrypted ciphertext becomes 7269767679. The recipient knows that each letter corresponds to a two-digit number and gets

$$\begin{aligned}
72 &\rightarrow \text{H} \\
69 &\rightarrow \text{E} \\
76 &\rightarrow \text{L} \\
76 &\rightarrow \text{L} \\
79 &\rightarrow \text{O}.
\end{aligned}$$

The message `HELLO` is then decoded which is the same as the integer that represents the original plaintext.

2.2.2.2 Paillier Cryptosystem

Being a public key encryption scheme, the Paillier cryptosystem [116] is invented in 1999 by French mathematician Pascal Paillier. It is a probabilistic asymmetric algorithm and is based on decisional composite residuosity assumption[9]. The Paillier encryption scheme, in addition, is additive homomorphic, i.e., given only the encryption of m_1 and the encryption of m_2 with the public-key, the encryption of $m_1 + m_2$ can be computed. The scheme (Figure 2.4) works as follows.

Key Generation:

1. Choose randomly two large prime numbers p and q such that pq and $\varphi(pq)$ are relatively prime, i.e., $gcd(pq, \varphi(pq)) = 1$ where $\varphi(.)$ is Euler's totient function (or phi function) with $\varphi(pq) = (p-1)(q-1)$.
2. Compute $n = pq$ and $\lambda = lcm(p-1, q-1)$.
3. Select a random integer g where $g \in \mathbb{Z}^*_{n^2}$.
4. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse

$$\mu = (L(g^{\lambda} \mod n^2))^{-1} \mod n,$$

[9] *Decisional composite residuosity assumption*: Given an integer z and a composite number $n = pq$ for primes p and q. It is hard to decide whether z is an n-residue modulo n^2 or not.

Public key	:	(n, g)
Private key	:	(λ, μ)
Encryption	:	plaintext $m < n$ random value $r < n$ ciphertext $c = g^m.r^n \mod n^2$
Decryption	:	ciphertext $c < n^2$ plaintext $m = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n$ plaintext $m = L(c^\lambda \mod n^2).\mu \mod n$

Figure 2.4: Paillier Probabilistic Encryption Scheme [116]

where function L is defined as $L(u) = \frac{u-1}{n}$. Note that this quotient does not denote the modular multiplicative of inverse of n times $u-1$ but rather the quotient of $u-1$ divided by n, i.e., the largest integer value $v \geq 0$ to satisfy the relation $(u-1) \geq vn$.

Then the public key is (n, g) and the private key is (λ, μ).

Encryption:

1. Let m be a plaintext where $m \in \mathbb{Z}_n$.
2. Select a random value r where $r \in \mathbb{Z}_n^*$.
3. Compute ciphertext as $c = \mathcal{E}_{pk}(m, r) = g^m.r^n \mod n^2$ by using the recipient's public key pk.

Decryption:

1. Ciphertext $c \in \mathbb{Z}_{n^2}^*$.
2. Compute plaintext as $m = L(c^\lambda \mod n^2).\mu \mod n$ or more explicitly, $m = L(c^\lambda \mod n^2).L(g^\lambda \mod n^2)^{-1} \mod n$.

Example 2.2. Suppose a sender wants to scramble the plaintext `HELLO` using Paillier encryption. From the ASCII table (Table 2.1), the plaintext message `HELLO` is mapped to integers of the ASCII encodings as $m = 7269767679$.

Setup:

1. Let $p = 293$ and $q = 433$ be two distinct 9-bit-length primes with $gcd(pq, \varphi(pq)) = 1$ where $\varphi(pq) = (p-1)(q-1)$.
2. $n = pq = 293.433 = 126869$ and $\lambda = lcm(p-1, q-1) = lcm(292, 432) = 31536$.
3. Ensuring n divides the order of g, select $g = 6497955158 \in_R \mathbb{Z}^*_{n^2}$ where $n^2 = 16095743161$.
4. Then μ becomes
$$\begin{aligned} &= (L(g^{\lambda} \mod n^2))^{-1} \mod n \\ &= (L(6497955158^{31536} \mod 16095743161))^{-1} \mod 126869 \\ &= (L(3967320500))^{-1} \mod 126869 \\ &= 31271^{-1} \mod 126869 \\ &= 53022. \end{aligned}$$

Now, $(n, g) = (126869, 6497955158)$ becomes public parameters whilst the pair $(\lambda, \mu) = (31536, 53022)$ remains private. Note that, these keys are the recipient's keys.

Encryption:

1. The plaintext is $m = 7269767679$. Now, the sender divides this numeric plaintext into segments. Each segment must contain the largest possible number less than $n = 126869$. That is,
$$s_1 = 72697, s_2 = 67679.$$
Then, choosing $r = 7 < 126869 = n$ as the random value the sender gets
$$\begin{aligned} \mathcal{E}_{pk}(s_1) = \mathcal{E}_{pk}(72697) &= 6497955158^{72697}.7^{126869} \mod 16095743161 \\ &= 7115464588 \\ \mathcal{E}_{pk}(s_2) = \mathcal{E}_{pk}(67679) &= 6497955158^{67679}.7^{126869} \mod 16095743161 \\ &= 3008149340. \end{aligned}$$
2. The sender concatenates all the outputs and gets the ciphertext as
$$\mathcal{E}_{pk}(m) = 71154645883008149340$$
and sends this value to the recipient.

Decryption:

1. The recipient takes the ciphertext $c = 71154645883008149340$ which can be divided into segments each of which is less than $n^2 = 16095743161$ as
$$m_1 = 7115464588, m_2 = 3008149340.$$

Then, the recipient uses his/her private key $sk = (\lambda, \mu) = (31536, 53022)$ and gets

$$\begin{aligned}
\mathcal{D}_{sk}(m_1) &= \mathcal{D}_{sk}(7115464588) \\
&= \frac{L(7115464588^{31536} \mod 126869^2)}{L(6497955158^{31536} \mod 126869^2)} \mod 126869 \\
&= L(7115464588^{31536} \mod 126869^2).\mu \mod 126869 \\
&= L(7115464588^{31536} \mod 126869^2).53022 \mod 126869 \\
&= L(8772357006).53022 \mod 126869 \\
&= 69145.53022 \mod 126869 \\
&= 72697 \\
\mathcal{D}_{sk}(m_2) &= \mathcal{D}_{sk}(3008149340) \\
&= \frac{L(3008149340^{31536} \mod 126869^2)}{L(6497955158^{31536} \mod 126869^2)} \mod 126869 \\
&= L(3008149340^{31536} \mod 126869^2).\mu \mod 126869 \\
&= L(3008149340^{31536} \mod 126869^2).53022 \mod 126869 \\
&= L(11192383181).53022 \mod 126869 \\
&= 88220.53022 \mod 126869 \\
&= 67679
\end{aligned}$$

2. Concatenating all the outputs, decrypted ciphertext becomes 7269767679. The recipient knows that each letter corresponds to a two-digit number and gets

$$\begin{aligned}
72 &\rightarrow \text{H} \\
69 &\rightarrow \text{E} \\
76 &\rightarrow \text{L} \\
76 &\rightarrow \text{L} \\
79 &\rightarrow \text{O}.
\end{aligned}$$

The message `HELLO` is then decoded which is the same as the integer that represents the original plaintext.

2.2.3 Cryptographic Hash Functions

Cryptographic hash functions (hereafter, we use simply hash functions) play a basic role for many algorithms in modern cryptography. According to Menezes et al. [100], the *hash function*, Hash, maps binary strings of arbitrary finite length to binary strings of some fixed length, say n bits. In other words, this mechanism maps a large (practically infinite) domain to a fixed range, mathematically stated as for a domain D and range R with a hash function $\mathsf{Hash} : D \rightarrow R$ with $|D| > |R|$.

The data to be encoded by a hash function are often called a *message*, and the values returned by a hash function are called *hash values*[10].

The hash function is many-to-one and therefore possesses a compression property, however potential collision is inevitable. Actually, restricting Hash to a domain of t-bit inputs ($t > n$), if Hash were "random" in the sense that all outputs were essentially equally likely to, then about 2^{t-n} inputs would map to each output, and two randomly chosen inputs would yield the same output with probability 2^{-n} (independent of t) [100].

A hash function has the following two properties [100] one of which is discussed earlier.

- *Compression:* Hash maps an arbitrary finite bit-length input x to an output $\mathsf{Hash}(x)$ of fixed bit-length n.

- *Ease of computation:* Given a hash function Hash and an input x, the output $\mathsf{Hash}(x)$ is easy to compute.

In addition to those two properties, a hash function satisfies some non-invertibility such as *one-way or preimage resistance*, *second-preimage resistance* and *collision resistance* properties. These properties become an important component in the security of cryptographic algorithms. The definitions of those properties are in the following [100].

- *One-way or preimage resistance:* If we have a predetermined output, it is computationally infeasible to find an input which hashes that input to that output, i.e., given an output $\mathsf{Hash}(x)$ and a hash function Hash, it is computationally infeasible to find any preimage x' with $\mathsf{Hash}(x') = \mathsf{Hash}(x)$.

- *Second-preimage resistance:* If we have a specified input and its output, it is computationally infeasible to find a second input whose output is the same as the given output, i.e., given a preimage x, hash function Hash, output $\mathsf{Hash}(x)$ and preimage property, it is computationally infeasible to find a second-preimage $x' \neq x$ such that $\mathsf{Hash}(x') = \mathsf{Hash}(x)$.

- *Collision resistance:* It is computationally infeasible to find any two distinct inputs $x \neq x'$ with $\mathsf{Hash}(x) = \mathsf{Hash}(x')$. In another words two distinct inputs should not hash into same outputs. Collision resistance property offers strongest security including preimage and second-preimage resistance.

Cryptographic hash functions can be used for encryptions and message authentication codes (MACs) (also called keyed hash functions). Because of the compression property, they can also be used to make digital signatures more efficient [53].

[10] Instead of *hash value*, other phrases such as *message digest*, *hash code*, *hash sum* or *checksum* are also used.

There are so many cryptographic hash functions most of which are vulnerable and unfortunately unused. The most popular cryptographic hash functions are constructed using the Merkle-Damgård iterative structure of hash function. Most of the using hash functions take this form, e.g., SHA-1 and MD5 [122]. Other hash function examples are HAVAL [152], RIPEMD[11], RIPEMD-128/256, RIPEMD-160 [52], RIPEMD-320, SHA-0[12], SHA-256/224, SHA-512/384, SHA-3[13] (originally known as Keccak[14]) [15] and WHIRLPOOL [9].

2.2.4 Threshold Cryptography

Sometimes accessing to precious items is controlled by a single part only. For instance, two keys are required to open a personal safe at a bank; one of the keys is kept by the bank and the other is kept by the bank customer or owner of the safe. In a similar way, if we think of a cryptographic key (secret key) instead of a safe key, single ownership of the secret key is also undesirable. Thus, the ownership (i.e., knowledge) of a secret key is distributed between several parties [127].

Threshold cryptography, also called as group-oriented cryptography, contains techniques to distribute basic cryptographic schemes between several parties [127], e.g., in a threshold version of a digital signature scheme, the private key is shared between n parties such that each subset of t parties (or more) is able to issue signatures, while subsets of less than t parties cannot produce valid signatures.

2.2.4.1 Secret Sharing

Secret sharing schemes are the basis of threshold cryptography [127]. In cryptography, *secret sharing* is a method for distributing a secret among a group of participants, each of which takes a share of the secret. The simplest form of a secret sharing scheme is one that requires all n participants to be present in order to reconstruct the secret while keeping it hidden for any smaller group. Note that, in a protocol a *dealer* D is the one who shares a secret s such that each *participant* P_i takes a share s_i, $1 \leq i \leq n$ of s. More complex schemes also exist, schemes that require a threshold number of people to cooperate in order to reconstruct the secret and even more flexible schemes that allow predefined groups of people to recover the secret [106].

In *additive secret sharing* the reconstruction of a secret s, is trivial; simply all of the shares, $s_1, \ldots, s_n$ are added together. In this scheme, initially the first $n-1$

[11] RIPEMD is not used widely anymore since a strengthened version of it has been released.

[12] SHA-0 is not used widely anymore since a strengthened version of it has been released.

[13] SHA-2 is not replaced by SHA-3 since no significant attack has been shown for SHA-2. The successful attacks on MD5, SHA-0 and theoretical attacks on SHA-1 made National Institute of Standards and Technology (NIST) require an alternative, dissimilar cryptographic hash algorithm, which became SHA-3.

[14] On October 2, 2012, *Keccak* was the winner of the NIST hash function competition [21].

shares $s_1, \ldots, s_{n-1}$ are generated at random by a trusted party and the last share s_n is set as follows

$$s_n = s - \sum_{i=1}^{n-1} s_i.$$

The secret is then recovered as

$$s = \sum_{i=1}^{n} s_i.$$

This scheme requires all n participants to contribute their shares in order to reconstruct the secret s. If one or more of the participants are missing, no information about the original secret can be recovered; such a scheme is known as a *perfect secret sharing scheme* [106].

In 1979, both Shamir [131] and Blakley [16] presented simple, tough powerful secret sharing schemes that allowed a t-threshold of n people, where $t \leq n$, to reconstruct the secret [106]. Each of n people is given a number of share, and any group of t or more shares such that $t \leq n$ together can open the secret but no group of less than t shares can. In the most general case, such a system is called a (t, n)-*threshold scheme*. If the knowledge of $t - 1$ or fewer shares gives no information about the secret s then that threshold scheme is called *perfect*.

In Chapter 4, our proposed solution is based on $(2, 2)$-threshold encryption. As discussed earlier, there are three parties in our proposed protocol and each of them has their own secret key. But each of the secret keys owned by the primary dealers is shared among the related primary dealer and the Treasury. This means that when there is a need for the secret key of the intended primary dealer, the Treasury and that primary dealer must contribute their shares to reconstruct the secret key. Thus, individual shares are of no use.

2.2.4.2 Shamir's (t, n)-Threshold Scheme

Shamir's scheme is an example of a perfect (t, n)-threshold scheme in which a classical algorithm called *Lagrange interpolation* is used. We first introduce Lagrange interpolation with a theorem without giving the proof. For the proof, Tavernini's lecture notes [136] can be seen.

Theorem 2.1. ***(Lagrange interpolation)*** *Given t distinct points (x_i, y_i) which are of the form $(x_i, f(x_i))$ with $f(x)$ being a polynomial of degree less that t, then $f(x)$ can be determined by the Lagrange interpolation formula as*

$$f(x) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t, j \neq t} \frac{x - x_j}{x_i - x_j}.$$

Shamir's threshold scheme is defined for a secret $s \in \mathbb{Z}/p\mathbb{Z} = \mathbb{F}_p$ with prime p, by setting $a_0 = s$ and choosing $a_1, \ldots, a_{t-1}$ at random in $\mathbb{F}_p$. A trusted party computes $f(x_i)$ for all $1 \leq i \leq n$, where

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}.$$

The shares $(x_i, f(x_i))$ are distributed to n distinct parties. Since the secret is the constant term, i.e., $s = a_0 = f(0)$, the secret is recovered from any t shares $(x_i, f(x_i))$, for $I \subset 1, \ldots, n$ by

$$s = \sum_{i \in I} c_i f(x_i), \text{ where each } c_i = \prod_{j \in I, j \neq i} \frac{x_i}{x_j - x_i}.$$

The following example illustrates the basic idea [91].

Example 2.3. Set $p = 31$ and let the threshold be $t = 3$ and the secret be $7 \in \mathbb{F}_{31}$. We choose the coefficients at random $a_1 = 19$ and $a_2 = 21$ in $\mathbb{F}_{31}$. Then the polynomial becomes $f(x) = 7 + 19x + 21x^2$. Being the trusted party, we can generate as many shares as we like as in the following.

$$\begin{array}{lll} (1, f(1)) = (1, 16) & (2, f(2)) = (2, 5) & (3, f(3)) = (3, 5) \\ (4, f(4)) = (4, 16) & (5, f(5)) = (5, 7) & (6, f(6)) = (6, 9) \\ (7, f(7)) = (7, 22) & (8, f(8)) = (8, 15) & (9, f(9)) = (9, 19) \end{array}$$

These shares are distributed to different parties who are called holders of the share recipients, and then the original polynomial $f(x)$ is killed. The secret can be recovered using the Lagrange interpolation formula as

$$f(x) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t, i \neq j} \frac{x - x_j}{x_i - x_j} \quad \Longrightarrow \quad f(0) = \sum_{i=1}^{t} y_i \prod_{1 \leq j \leq t, i \neq j} \frac{x_j}{x_j - x_i}$$

taking any t shares $(x_1, y_1), \ldots, (x_t, y_t)$. Let us take the first three shares $(1, 16)$, $(2, 5)$ and $(3, 5)$. Then we can compute the constant term

$$\begin{aligned} f(0) &= \frac{16.2.3}{(1-2)(1-3)} + \frac{5.1.3}{(2-1)(2-3)} + \frac{5.1.2}{(3-1)(3-2)} \\ &= 3.2^{-1} + 15.(-1) + 10.2^{-1} \quad \text{in } \mathbb{F}_{31} \\ &= 3.16 + 15.(-1) + 10.16 \quad \text{in } \mathbb{F}_{31} \\ &= 17 + 16 + 5 \quad \text{in } \mathbb{F}_{31} \\ &= 7. \end{aligned}$$

The result would be the same if we choose another three shares.

2.2.4.3 Verifiable Secret Sharing

Basic secret sharing schemes are resisted passive attacks only. This means that the security of such schemes requires all parties involved in the protocol are assumed to follow the protocol every time exactly as directed by the scheme. If a set of participants in the distribution protocol acts honestly, then none of them can gather any information on the secret. However, in many applications, it is needed to resist active attacks. This is achieved by a *verifiable secret sharing* (VSS) scheme. By VSS, it is aimed schemes to be resisted the following two types of active attacks [127]:

- During the distribution phase, a dealer may send incorrect shares to some or all of the participants, and
- During the reconstruction phase, participants may submit incorrect shares.

Unambiguously, Shamir's scheme is not a VSS scheme, because it does not prevent either of these active attacks. As an example to VSS, Feldman's VSS or Pedersen's VSS can be given [127].

2.2.4.4 Threshold Paillier Cryptosystem

A function sharing scheme for the Paillier cryptosystem was proposed by Fouque et al. [63] based on Shamir's secret sharing. Here is the key generation, encryption, share decryption and combining algorithms for the threshold version of Paillier cryptosystem.

Key Generation:

1. Choose two integers p' and q' such that $p = 2p' + 1$ and $q = 2q' + 1$ are primes and such that $m = p'q'$, $n = pq$ and $gcd(n, \varphi(n)) = 1$.
2. Let a, b, β be elements randomly chosen from $\mathbb{Z}_n^*$.
3. Set $g = (1 + n)^a b^n \mod n^2$.
4. Set $\theta = L(g^{m\beta}) = am\beta \mod n$ where $L(u) = \frac{u-1}{n}$.
5. The public key is $pk = (n, g, \theta)$ and the secret key is $sk = m\beta$.

Share Initialization:

1. The secret key $sk = m\beta$ is shared between the parties by using the Shamir secret scheme, setting $a_0 = m\beta$ and choosing $a_i \in \{0, \ldots, nm - 1\}$ at random for $i = 1, \ldots, t - 1$ such that $f(x) = \sum_{i=1}^{t-1} a_i x^i$.

2. The share s_i of the i^{th} party P_i which is $s_i = f(i) \mod nm$ is sent to P_i.
3. Choose a random v from the subgroup of squares of $\mathbb{Z}^*_{n^2}$.
4. Make $v^{\Delta s_i} \mod n^2$ public where $\Delta = \ell!$ with ℓ being the number of parties.

Encryption:

To encrypt a message M with a public key pk, randomly pick $r \in \mathbb{Z}^*_n$ and compute the ciphertext

$$c = g^M r^n \mod n^2.$$

Decryption Share Generation:

The i^{th} party P_i computes the decryption share $c_i = c^{2\Delta s_i} \mod n^2$ using his/her secret share s_i. He/she makes a proof of correct decryption which assures that $c^{4\Delta} \mod n^2$ and $v^{\Delta} \mod n^2$ have been raised to the same power s_i in order to get ${c_i}^2$ and v_i, i.e., $\log_{c^{4\Delta}} c_i = \log_{v^{\Delta}} v_i$.

Decryption Share Reconstruction:

If less than t decryption shares have valid proofs of correctness, the algorithm fails. Otherwise, let S be a set of $t+1$ valid shares and compute the plaintext

$$M = L\left(\prod_{j \in S} {c_j}^{2\mu^S_{0,j}} \mod n^2\right) \frac{1}{4\Delta^2\theta} \mod n$$

where $L(u) = \frac{u-1}{n}$ and $\mu^S_{0,j} = \Delta \prod_{j' \in S - \{j\}} \frac{j'}{j'-j} \in \mathbb{Z}$.

The details are presented in [63] and [76].

2.2.5 Homomorphic Encryption

Homomorphic encryption is a form of encryption in which all the computations are carried out using the ciphertexts only. The output is the ciphertext of the result of operations performed on the plaintext [76].

Homomorphic encryption schemes are malleable[15] by design which can be used to create for example secure voting systems, private information retrieval schemes. They also enable extensive use of cloud computing by ensuring the confidentiality of processed data [76]. For any two plaintexts m_1, m_2 and an encryption algorithm $\mathcal{E}_{pk}$, if the equality

$$\mathcal{E}_{pk}(m_1) \cdot \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 + m_2)$$

[15] An encryption algorithm is *malleable* if it is possible for an adversary to generate a ciphertext from known ciphertexts where the generated ciphertext has a valid decryption. That is, given an encryption of a plaintext p, it is possible to generate another ciphertext which decrypts to $f(p)$, for a known function f, without necessarily knowing or learning p.

is satisfied then we say the system is *additive homomorphic*. ElGamal [57], Paillier [116] and Benaloh [32] schemes are examples for additive homomorphic encryption. It is *multiplicative homomorphic* if

$$\mathcal{E}_{pk}(m_1) \cdot \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 \cdot m_2)$$

is satisfied. Multiplicative ElGamal and textbook RSA schemes are examples for multiplicative homomorphic encryption. RSA-OAEP[16], on the contrary, is not homomorphic because of randomness that is run in the algorithm which can be used to convert a deterministic encryption scheme (e.g., plain RSA) into a probabilistic scheme[17].

The schemes such as ElGamal, textbook RSA, Paillier and Benaloh allows homomorphic computation of only one operation (either addition or multiplication) on plaintexts. A cryptosystem supporting both addition and multiplication operations with preserving the ring structure of the plaintexts is known as *fully homomorphic encryption*. Any circuit can be homomorphically evaluated by using such an encryption which is a much more powerful technique. Also programs can be constructed effectively running on encryptions of their inputs to produce encryptions of their output. Since those programs do not decrypt its input, they can be run by an untrusted party without revealing the inputs and internal state. These bring on great practical implications in the outsourcing of private computations, e.g., in the context of cloud computing [101]. Craig Gentry [68] using lattice-based cryptography outlined the first fully homomorphic encryption scheme in 2009 [33]. His scheme supports evaluations of arbitrary depth circuits. However, current implementations are far from practicality.

Additive Homomorphic Property of Paillier Encryption Scheme:

Let m_1 and m_2 be two plaintexts to be encrypted by Paillier algorithm and let the the corresponding ciphertexts be c_1 and c_2 respectively. If we consider $m_1 + m_2$ and encrypt this sum then we get the encryption of $c_1.c_2$ which shows that the Paillier Encryption is *additive homomorphic*. This is because of using exponentiation in the encryption algorithm.

---

[16] In cryptography, *Optimal Asymmetric Encryption Padding* (OAEP) is a padding scheme. It is often used with RSA encryption. OAEP was first introduced by Bellare and Rogaway [11] and subsequently standardized in PKCS #1 version 2.1 (RFC 3447).

[17] In plain RSA, semantic security fails since encryption is deterministic. Even worse, under a chosen-ciphertext attack (CCA attack), the attacker can fully decrypt a challenge ciphertext using the homomorphic property of plain RSA: $\mathcal{E}_{pk}(m_1) \cdot \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 \cdot m_2)$. That is, an attacker tries to learn the decryption of a ciphertext $c = m^e \mod n$ and he/she may ask the owner of the private key to decrypt an unsuspicious-looking ciphertext $c' = cr^e \mod n$ for some value r chosen by his/her. Because of the multiplicative property of plain RSA, $c' = cr^e \mod n = (m^e \mod n)(r^e \mod n) = (mr)^e \mod n$. Hence, if the attacker becomes successful in this attack, he/she will learn $mr \mod n$ from which he/she can derive the message m by multiplying mr with $r^{-1} \mod n$. *To overcome this attack*, the plaintext is randomly padded before encryption process in practical RSA-based cryptosystems. Adding good padding (OAEP) will make RSA semantically secure as desired and randomizes the ciphertext, but then *eliminates the homomorphic property* [120].

$$\begin{aligned}\mathcal{E}_{pk}(m_1, r_1) \cdot \mathcal{E}_{pk}(m_2, r_2) &= (g_1^m.r_1^n \mod n^2).(g_2^m.r_2^n \mod n^2) \\ &= (g_1^m.r_1^n).(g_2^m.r_2^n) \mod n^2 \\ &= (g^{m_1+m_2}.(r_1.r_2)^n \mod n^2 \\ &= \mathcal{E}_{pk}(m_1 + m_2, r_1.r_2) \\ &= \mathcal{E}_{pk}(m_1 + m_2, R).\end{aligned}$$

This property with the following property

$$\mathcal{E}_{pk}(k.m) = \mathcal{E}_{pk}(m)^k$$

together are known to be particularly appreciated in the design of voting protocols, threshold cryptosystems, watermarking, private information retrieval, secret sharing schemes and sharing of DSA signatures [116].

Example 2.4. Considering the same constraints of the previous example, let $m_1 = 8$ and $m_2 = 9$ be two plaintexts and $r_1 = 7$ and $r_2 = 11$ be the corresponding randoms. Then we have the following calculations showing the additive homomorphic property of Paillier encryption scheme.

$$\begin{aligned}\mathcal{E}_{pk}(m_1, r_1) \cdot \mathcal{E}_{pk}(m_2, r_2) &= \mathcal{E}_{pk}(8, 7) \cdot \mathcal{E}_{pk}(9, 11) \\ &= (6497955158^8.7^{126869}).(6497955158^9.11^{126869}) \mod 126869^2 \\ &= 6075462831.4638741447 \mod 126869^2 \\ c &= 4029386836.\end{aligned}$$

$$\begin{aligned}\mathcal{D}_{sk}(4029386836) &= L(c^\lambda \mod n^2).\mu \mod n \\ &= L(4029386836^{31536} \mod 126869^2).53022 \mod 126869 \\ &= L(3061475840).53022 \mod 126869 \\ &= 24131.53022 \mod 126869 \\ m &= 17 \\ &= 8 + 9 \\ &= m_1 + m_2.\end{aligned}$$

In our proposed model, we have summations on encrypted values (see Chapter 4). Since the values we use are in Paillier encrypted form then we product all those encrypted values to get the encryption of the sum owing to additive homomorphic property of the Paillier encryption scheme. Then the desired result is obtained by decrypting the encrypted sum value.

2.2.6 Digital Signature

The conventional handwritten signature on a document is used to confirm that the signer is responsible for the content of the document. On the other hand, in digital medium, we need to have a way of signing messages digitally which is functionally equivalent to the handwritten signature, but which is at least as resistant to forgery as its physical counterpart [31]. In 1976, in their famous paper [51], Diffie and Hellman firstly described the notion of a digital signature scheme. They only conjectured in their paper that such schemes existed. The first digital signature scheme was constructed by Rivest, Shamir and Adleman [123]. In their paper [123], they also proposed the first public key cryptosystem and paved the way for further study of cryptography [98]. Here is the formal definition of *digital signature scheme*.

Definition 2.3. A *digital signature scheme* is a five-tuple $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$ satisfying the following properties (see also Figure 2.5):

1. $\mathcal{P}$ is a finite set of possible messages,
2. $\mathcal{A}$ is a finite set of possible signatures,
3. $\mathcal{K}$ is the key space, a finite set of possible keys,
4. For each $K \in \mathcal{K}$ there exists a signature algorithm $\mathsf{Sign}_K \in \mathcal{S}$ and a verification algorithm $\mathsf{Ver}_K \in \mathcal{V}$ such that
   - $\mathsf{Sign}_K : \mathcal{P} \to \mathcal{A}$
   - $\mathsf{Ver}_K : \mathcal{P} \times \mathcal{A} \to \{\text{accept, not accept}\}$
   - For all $x \in \mathcal{P}$ and $y \in \mathcal{A}$

$$\mathsf{Ver}_K(x, y) = \begin{cases} \text{accept} & \text{if } y = \mathsf{Sign}_K(x) \\ \text{not accept} & \text{if } y \neq \mathsf{Sign}_K(x). \end{cases}$$

Here, the pair $(x, y) \in \mathcal{P} \times \mathcal{A}$ is called a *signed message*.

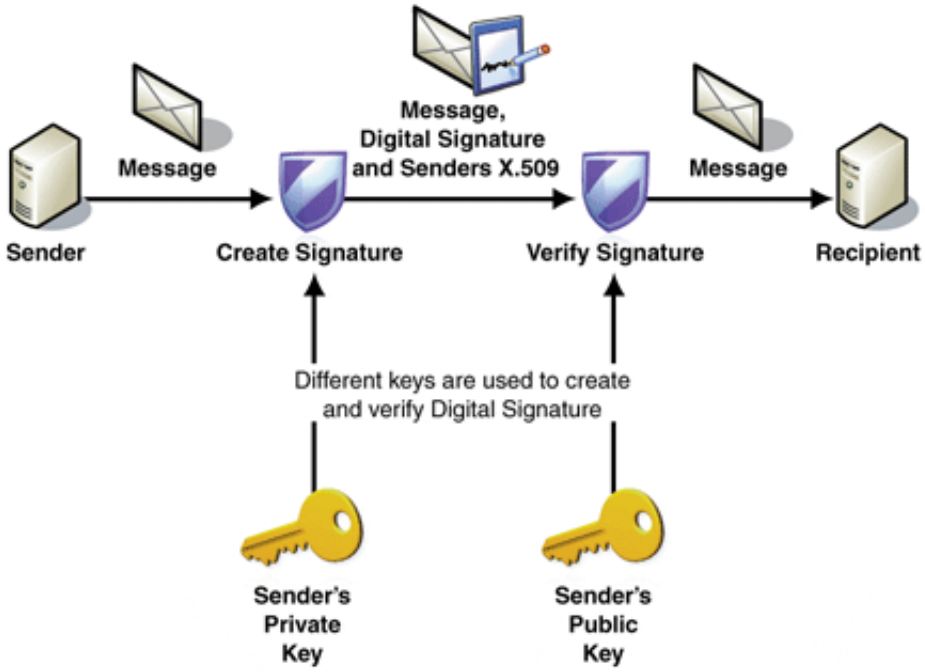

Figure 2.5: Creation and Verification of a Digital Signature [103]

Digital signatures have the same functions with the handwritten signatures. Whence, simiilar to a handwritten signature, a digital signature must be

- *message-dependent*, and

- *signer-dependent.*

The *message-dependency* means that the signature is *not reusable*. Thus, the signature is not valid for another document or for a modified initial document implying that the signature guarantees the *integrity* of the signed document. The *signer-dependency*, on the other hand, means that the signature is *unforgeable*. That is, nobody except the signer can sign a document. Hence, the recipient is satisfied that the signer intentionally signed the document implying that the signature satisfies the *authenticity* of the signer. Finally, these two properties satisfies the *non-repudiation* of the signature meaning that the signer cannot later claim that he/she did not sign the message [105].

Digital signatures use operations which are dependent on public key cryptography algorithms. NIST announced the approval of FIPS 186-4, *The Digital Signature Standard (DSS)*, issued on July 2013 as a standard [109]. Currently, there are three public key algorithms that are approved by Federal Information Processing Standards (FIPS) for purposes of generating and verifying digital signatures [140]. These are:

- DSA: Digital Signature Algorithm, specified in FIPS 186-4,
- ECDSA: Elliptic Curve DSA, specified in ANS X9.62,
- RSA: Specified in ANS X9.31 and PKCS #1.

Before constructing a digital signature, a signing *certificate* is needed for proving identity. When one sends a digitally-signed document or message, he/she also sends his/her certificate and public key. Certificates are issued and managed by an authority called as a *certification authority* (CA)[18] and like a driver's license, can be revoked. A certificate is generally valid for three years (this duration may increase if the length of public and private keys are increased). After the validity expired, the signer must renew, or get a new, signing certificate to establish identity.

When a sender wants to construct a digital signature, he/she first calculates hash value of the document or message to be sent, say of m. He/she then encrypts the message digest, $\mathsf{Hash}(m)$, using his/her private key (sk) to generate the digital signature, which is appended to or embedded within the message. Once the encrypted message is received by a recipient, he/she decrypts it by using the sender's public key (pk). The recipient can then calculate the hash of the original message and compare it with the hashed value included in the signature to verify the sender's identity. Non-repudiation is guaranteed by the fact that the sender's public key has itself been digitally signed by the CA that issued it [140].

To make the assurances of authenticity, integrity and non-repudiation as mentioned above, the content creator must digitally sign the content by using a signature that satisfies the following criteria [104]:

[18] A *certificate authority (CA)* works like a notary public. It issues digital certificates, signs those certificates to verify their validity and tracks which certificates have been revoked or have expired.

- The digital signature is valid.
- The certificate associated with the digital signature is valid.
- The signing organization, known as the *publisher*, is trusted (note that, signed documents, which have a valid time stamp, are considered to have valid signatures, regardless of the age of the signing certificate).
- The certificate associated with the digital signature is issued to the signing publisher by a reputable CA.

Currently, the most popular signature algorithm is RSA with SHA-1. It uses keys of 1024 or 2048 bits long [78]. DSA, ECDSA, ElGamal signature scheme, Schnorr signature, Pointcheval-Stern signature algorithm and Rabin signature algorithm are other examples of signature schemes.

2.2.6.1 RSA Signature Scheme

Using the same notations of RSA which are described in Section 2.2.2.1, we have the fact that the RSA public key (n, e) is used to encrypt messages or to verify signatures, and the RSA private key (n, d) is used to decrypt encrypted messages or to sign messages. As defined in [105], to create a digital signature s for a message m or $f(m)$, where f is a hash or redundancy function, one uses his/her private key (n, d) to obtain s by exponentiating:

$$s = m^d \mod n \quad \text{or} \quad s = f(m)^d \mod n.$$

To verify the signature s, one uses the public key of the signer. He/she exponentiates and checks that the message m or $f(m)$ is recovered:

$$m = s^e \mod n \quad \text{or} \quad f(m) = s^e \mod n.$$

2.2.7 Secure Multi-Party Computation

Today's communication facilities allow accessing to almost any imaginable resources and connecting to any person easily. At the same time, the underlying technology only provides a "best effort" service [92]. That is, when for example Alice asks for something, it will probably be done, e.g. Alice wants to send a message to Bob. However, this message not only may be lost, it may also be read and, more importantly, modified by an attacker, while it is transmitted. Thus, protection against eavesdropping on the legitimate communication becomes the most common issue to be considered.

Assume we have solved the above problem and Alice is completely satisfied that her communication with Bob, Carmen, and others is private and authentic. This is not alone enough to realize a secure communication. For example, imagining a case where Alice communicates with Bob, but she does not fully trust him. This

may happen in many cases where the participants may have common interests such as contract signing or buy/sell transactions. Securing the communication channel cannot provide any guarantee that Bob does not cheat. Fortunately, a study of secure *multi-party computation* (MPC), which began in 1980's, emerged from the need to not only communicate, but also to *compute* securely. It addresses the problem of providing security against cheating *participants* of the computation [92]. Note that, MPC is a generalization of *secure function evaluation* and in an MPC protocol the parties have inputs and produce outputs several times during the computations [77].

If two parties exist, this case is called secure *two-party computation* (2PC). Two well-known examples of 2PC problems are as follows.

- *Yao's Millionaire Problem* [148]: Consider two millionaires want to reveal who is richer in such a way that their own fortunes remain private, but the correctness of the output can be checked by both of them. The function computed in this case is a simple comparison between two integers:

  $$f(x_1, x_2) = x_1 \overset{?}{>} x_2.$$

  If the result is "Yes", then the first millionaire is richer and it will be known that the second person has fewer millions than the first one, but this should be all the information they learn about the fortunes.

- *Love Game* (also known as dating problem) [30]: Alice and Bob are very shy and they want to learn whether they are interested in each other. But, if only Alice is interested, then she does not want to let Bob know that she is interested in him. The same holds if only Bob is interested, i.e., if a party is not interested then it should not be possible to learn the other party's decision.

As seen from the above examples 2PC and MPC are the problems of evaluating a function of two or more parties' secret inputs. In both computation techniques, each party finally holds a share of the function output and no more else is revealed, except what is implied by the party's own inputs and outputs. Secure MPC problem was first introduced by Yao [148] and extended by Goldreich et al. [71], and many others. It is easy to solve these kinds of problems if it is assumed that a *trusted third party* (TTP) which collects the inputs, evaluates the function and distributes the result to all participating parties exists. If no TTP is available, conversely, the problem becomes very difficult and parties can misbehave arbitrarily, i.e., they can send false messages or fail to send messages at all. Even then, in a secure MPC, the parties must be ensured that the protocol computes the function correctly and securely "as if" a TTP is available.

Let us define the problem we have to solve more clearly. Let $P_1, \ldots, P_n$ be the *parties* who participated into the computation. Each party P_i have a secret input x_i, and they agree on some function f that takes n inputs. Their goal is to compute $y = f(x_1, \ldots, x_n)$ *securely* while making sure that the following conditions are satisfied:

- *Correctness.* With this condition, it is ensured that the output of the function f that the parties receive is correct. In other words, no party can affect the output of the computation by changing its own input.

- *Privacy.* With this condition, no party should learn anything more than the output of the function f and his/her own input, i.e., no party can obtain information about the honest party's input.

- *Fairness.* With this condition, it is ensured that a corrupted party should receive his/her output if and only if the honest party also receives his/her output. In general, if a party is corrupted, then the adversary may learn the output before the honest party learn it and then may decide to abort the protocol. This may result in an unfair situation.

Secure MPC suggests solutions to various real-life problems. Consider n distrustful entities each of whom has a secret input x_i, $i = 1, \ldots, n$ and each of whom wants to evaluate the value $f_i(x_1, \ldots, x_n)$ where f_i's are n-input functions of the i^{th} party P_i. However, this evaluation should not leak any information about the other $n-1$ inputs to the related party. As an example, one may think of x_i as a number, namely P_i's bid in an *auction*, and f as a function to be evaluated,

$$f(x_1, \ldots, x_n) = (x_j, j)$$

where $x_j \geq x_i$ with $i = 1, \ldots, n$, i.e., f outputs the highest bid x_j, and the identity j of the corresponding bidder [36]. If we do not want the winner to pay as specified in his/her own bid, but the bid of the second highest bidder, we simply change x_j to be this value, which is again a well-defined function of the inputs. This would give us a function implementing a so-called second price auction [37].

Another practical example is *electronic voting scheme* with each x_i being a secret input (i.e., vote) of voter V_i and

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i$$

being the function to be evaluated securely. For simplicity, consider the case where n voters $V_1, \ldots, V_n$ want to vote on a plebiscite[19]. We can represent the votes such that $x_i = 0$ means "no" and $x_i = 1$ means "yes". If we can compute the sum of all x_i's securely, $\sum_{i=1}^{n} x_i$ is indeed the result of the vote, namely the number of *yes*-votes. Furthermore, if the computation is secure, no information is leaked other than $\sum_{i=1}^{n} x_i$, in particular, no information is revealed on how a particular voter voted [37].

Contract signing [13, 107] is another real-life example. In a contract signing protocol, the users have agreed on a contract over the network and want to obtain each other's signature on it. The signature exchange must be "simultaneous".

[19] A *plebiscite* is not an election since there are no candidates. Rather, people vote *yes* or *no* on a proposition.

Thus, *fairness* is an important property of contract signing protocols. That is no participant receives a contract with signatures of others, and no participant sends a contract having his/her signature on it clearly. When the signatures are completed then it becomes a valid contract.

The last practical example we give is *private information retrieval* (PIR) schemes. PIR protocol allows a user to retrieve an item from a server in possession of a database without revealing which item he/she is retrieving. PIR is a weaker version of 1-out-of-n oblivious transfer[20], where it is also required that the user should not get information about other database items.

MPC has been studied since the 1980's [71, 14, 29]. Until recently, mostly academic works[21] have been studied, because the related protocols add a fair amount of computational and network communication overhead [19]. Thus, until recently, most of the mechanisms developed so far have not been implemented, and applying on real-life is very limited [129]. However, many business applications could use and benefit from secure computation and in recent years, many MPC projects started to use in practice [12, 17, 20, 75]. The mechanism, used by Danish farmers to buy and sell contracts for sugar beet production on a nation-wide market, proposed in [20] is the first large-scale and practical application of MPC. In Estonia, a secure system for jointly collecting and analyzing financial data for a consortium of ICT[22] companies was developed and in this system secret sharing and secure MPC techniques were used. This was the first time where the actual secure multi-party function evaluation was performed over internet using real data. The details are presented in [19] and [17].

As we present in this thesis, our work also employs a secure MPC protocol to accomplish a secure auction scheme. In Chapter 4, we propose a protocol in the presence of corrupted parties under the assumption that the Treasury do not collude with the Central Bank, which are the two of the parties in our proposed protocol. In this model, we use secure sorting with secure comparison as an application of secure MPC by utilizing the recent work about the secure comparison in [145].

[20] An *oblivious transfer* protocol (often abbreviated OT) is a type of protocol in which there are a sender, a receiver and many pieces of information. In this protocol, the sender transfers one of the pieces of information to the receiver, where the receiver remains oblivious as to what piece has been transferred. A more useful form of OT called *1-2 OT* or *1-out-of-2 OT* was proposed later by Shimon Even, Oded Goldreich, and Abraham Lempel [59], in order to build protocols for secure MPC. Considering the PIR scheme, it can be generalized to 1-*out-of-n OT* where the user gets exactly one item or element without the server getting to know which item or element was queried, and without the user knowing anything about the other items or elements that were not retrieved.

[21] For an overview of the known theoretical results, [35] can be seen.

[22] ICT: Information and Communication Technologies

CHAPTER 3

ELECTRONIC AUCTION & DOMESTIC BORROWING

" If you haven't found it yet, keep looking. Don't settle. As with all matters of the heart, you'll know when you find it. And, like any great relationship, it just gets better and better as the years roll on."

— Steve Jobs

Auctions have been recorded as far back as 500 B.C. given by the Greek historian Herodotus, who described the sale of women to be wives in Babylonia. Beautiful maidens engendered lively bidding, but owners of the less comely women had to add a dowry or other monetary offer to make the sale [23]. The samples of so called negative-price auctions were encountered in these periods. Later on, auctions were used in many communities and civilizations for settling debt problems, for commercial trading, and for selling and buying goods or services.

Being occurred significant developments and improvements in the application and style of auctions, as a matter of course, different auction categories and schemes came into view. In this chapter, we discuss the most famous ones of them. We also mention about some selected Treasury auctions. The Turkish Treasury auction system, which is the most commonly used method for domestic borrowing [93] in Turkey, is also described in details, and an example of the current Turkish Treasury auction mechanism is outlined for ease of understanding at the end of the chapter. In the next chapter, we will use this model in order to construct a new and secure auction mechanism.

3.1 Types of Auctions

In this section, we categorize the auctions and present different existing schemes. We will later compare these auction solutions for domestic borrowing case.

Auctions are categorized in many ways according to the number of participants joined, properties of items auctioned, or price determination rules. As an illustration, our proposed solution is *single round sealed-bid multi-unit simultaneous*

dependent demand auction type with discriminatory price. We explain these terms as follows.

Single round auctions have a natural time requirement for the bids, i.e., all bids should be submitted by a certain deadline, after which the seller starts evaluating the outcome of the auction and rejects all further incoming bids.

Auctions in general can differ in the number of participants as illustrated in Figure 3.1.

- *Supply auction*,
- *Demand auction*,
- *Double auction*.

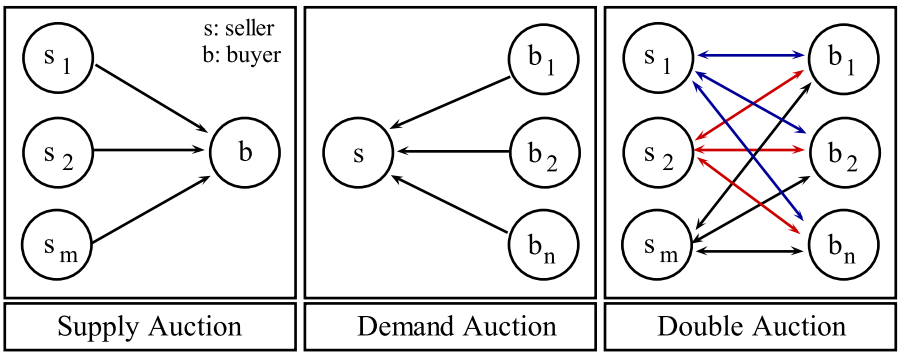

Figure 3.1: Auctions According to Number of Participants

In a *supply auction*, m sellers offer a good or an item that a buyer requests, e.g., government procurements. In a *demand auction*, n buyers bid for a good or item being sold. In a *double auction*, there is a trading process which contains both buying and selling goods or items where potential n buyers submit their bids and potential m sellers simultaneously submit their ask prices. The double auction is a stylized representation of organized exchanges like stock exchanges and commodity markets [99], e.g., NASDAQ, the New York Stock Exchange and the Chicago Board of Trade.

We can also divide auctions into three according to items[1].

1. **Single-Unit Auction.** In this type of auctions, only one item is sold. Methods utilizing this type include English, Dutch, and sealed-bid auctions [133]. As an example, if an auctioned item is a case of coffee mugs and bidders have to bid on the whole case, then this will be a *single unit auction*.

[1] A *heterogeneous item* is an item that performs a similar function but differs in another area. Let us consider a wristwatch as an example. One can have a wristwatch that tells time through an analog face or a digital face depending on what company makes them. On the other hand, a *homogeneous item* is one that is no matter from which company it is bought. It is essentially the same like whole milk. There are many dairies but all of the milk is the same.

2. **Multi-Unit Auction.** In this type of auctions, more than one homogeneous or identical items are being auctioned, rather than having separate auctions for each. Treasury bills auction is a good example for this kind of auction [70]. Large lots of refurbished items on eBay is another real-life example [117].

   Multi-unit auction literature starts with identifying three most common multi-unit formats. This classification is attributed to Weber [147].

   - *Simultaneous dependent auction*,
   - *Simultaneous independent auction*,
   - *Sequential auction*.

   In a *simultaneous dependent auction* the bidders are required to take a single action that determines both the allocation of the units and the payments to the seller. Weekly auction of US Treasury bills is an example of this kind of auction. A *simultaneous independent auction* is another type where the bidders must simultaneously act in several distinct auctions of individual items. In this type of auction, the outcome of each sale is independent of the outcomes of the others. This type is a special case of simultaneous dependent auction, which Weber chooses to view separately in [147]. The sale of mineral rights on federal land by the US Department of the Interior frequently takes this form [147]. Finally, a *sequential auction* is just what the name suggests; the sale of one item at a time, perhaps with the public release of information concerning the outcome of one round prior to the beginning of the next. Estate auctions at which a collection of objects –stamps, coins, antiques, or the like– are sold can be seen as examples of such sales. English and Dutch auctions are sequential auctions whereas first-price and Vickrey auctions can be categorized to simultaneous dependent sealed-bid auctions.

3. **Multi-Object Auction.** *Multi-object auction* is an auction in which heterogeneous or differentiated items are being auctioned. As an example we can consider a cattle auction in which each cattle would have different characteristics in terms of, say, weight, type and age.

There are mainly two rules used to determine the price [70].

1. *Uniform price rule.* Each winner of the auction pays the same price which is the highest price among the looser bids similar to the second price auction [70].

2. *Discriminatory price rule.* Each winner of the auction pays the price what he/she offered. This rule is also called pay-as bid or pay your bid auctions. It looks like the case of monopolist imposing discriminatory pricing [70].

For auctions with single object, four different formats have been studied extensively [70].

1. **English Auction.** In *English auction* (also called as ascending-bid or ascending open-cry auction), the price is consecutively increased until only one bidder remains. At any time during the English auction, the level of the current best bid is known by each bidder. This is the basic feature of the English auction. The auction ends when there is no one willing to bid further. At this point, the last offered price is the highest one and that is the auction price. Sometimes, the seller set a minimum sale price (the *reserve* price) before the auction. In this case if that pre-determined price is under the auction price, then the item remains unsold. To be held an English auction, at least two bidders are required. Antiques, artwork and secondhand goods, for example, are usually sold by English auction [99]. Sotheby's and Christie's which are auction houses use this method for auctioning fine art.

   The most common variation of English auction is the *Japanese auction*. In this auction method, the price is increased gradually while bidders quit the auction one after another. Bidders can observe whenever someone quits and none of them is able to re-enter the auction again [25]. Some other approaches for English auction can be found in [87, 94, 45, 6, 72].

2. **Dutch Auction.** *Dutch auction* (also called as descending-bid auction) is also an interactive auction which is the converse of the English auction. Namely, the auctioneer starts at a high price and continuously reduces it until one bidder expresses his/her willingness to pay [99]. It is called "Dutch auction" because flowers have long been sold in the Netherlands using this method [55].

   English auctions are more appropriate for unique items such as antiques and art works, and sealed-bid auctions are preferred if the buyers want to maintain some level of confidentiality. On the other hand, Dutch auctions work better for the disposal of perishable goods or the sale of products whose worth decreases in time such as produce, tobaccos, newspaper, seats on a flight or in a concert [96]. In particular, Dutch auctions have found widespread practical applications in cash management [4], stock repurchases or share buybacks [7, 67, 80], cloth sales [40], plant sales [85], vehicle slot sales [81], initial public offerings [124], fish sales in Israel [69] and tobacco sales in Canada [25]. Some studies on Dutch auctions can be found in [150, 151, 135, 39, 95, 96].

   Example 3.1. In the context of an initial public offering, investors submit their orders for the number of shares they want, and at what price. When there are enough investors willing to buy all the shares in the offering, the final price is formed at that level called *clearing price*. The investors who bid at or above this level get shares at that price, even if they would bid higher [46]. Let us assume Company ABC wants to sell 11 million shares using a Dutch auction, an investor typically opens an account with usually an investment bank and gets an access code or bidder identification code since Dutch auctions often occur online. During the bidding phase, investors specify how many shares they are willing to buy and the price they are willing to pay. The investment bank, who acts as the auctioneer, opens the auction by offering a high price for the security say, $30 per share

for this example. The bank then lowers the price to one level down, say, $23 per share, where some bids come in for totally 2,000,000 shares. The investment bank then lowers the price down again, this time to $22, and attracts 3,000,000 shares worth of bids. After lowering the price to $21, the investment bank gets 6,000,000 shares worth of bids; then the investment bank lowers the price to $20 and gets another 2,000,000 in bids before the auction ends (see Table 3.1).

Price	**Shares**	**Cumulative Shares**
$30	0	0
$23	2,000,000	2,000,000
$22	3,000,000	5,000,000
$21	6,000,000	11,000,000
$20	2,000,000	13,000,000

Table 3.1: Example Dutch Auction of Company ABC

After the auction is closed, the investment bank calculates the highest price at which all shares are sold. Here, the investment bank gets bids of totally 13 million shares, but the highest bids adding up to 11 million shares are the winning bids in the auction. The investment bank will then set the price equal to the lowest winning price bid on those 11 million shares (i.e., $21), and all the winning bidders will pay that price. Note that the price of $21 applies to all bidders, even the ones that bid $22 or $23.

3. **First-Price Sealed-Bid Auction.** With the *first-price sealed-bid auction*, potential buyers submit simultaneous sealed bids to the seller and the bidder that submitted the highest bid becomes the winner and pays the value that he/she specified in his/her bid. In an English auction, bidders have the chance of observing their rival's bids and accordingly, if they choose, revise their own bids; whereas in a sealed-bid auction, each bidder can submit only one bid and since they cannot see the bids of others they cannot adjust their own bids. This is the main difference between the first-price sealed-bid auction and the English auction. First-price sealed-bid auctions are used in the auctioning of mineral rights to US government-owned land and they also used in the sales of artwork and real estate [99]. US Treasury auctions used to be run this way too [90].

4. **Vickrey Auction.** In a *Vickrey auction* (also called as *second-price sealed-bid auction*), bidders submit simultaneous sealed bids to the sellers having been told that the highest bidder wins the item but pays a price equal not to his/her own bid but to the second highest bid[2]. The name *Vickrey* was

[2] If two or more bidders tie for the highest bid, either the winner is picked at random and has to pay the amount of his/her bid (because in this case it is equal to the second highest bid) [25] or the protocol yields no winners [24].

given in honor of William Spencer Vickrey[3]. Vickrey has shown that it is a strategy-proof mechanism to sell M indistinguishable units of the same item to the M highest bidders for the uniform price given by the $(M+1)$st highest bid [146]. In $(M+1)$st price auctions there are two phases: *bidding* and *opening* [3]. In bidding phase, the bidders submit their price offer for the auction in which there are M units of a single kind of goods. For this, each bidder determines and seals his/her price by an envelope for example, and puts it into the auctioneer's ballot box. In opening phase, i.e., just after all bidders have cast their sealed prices, the auctioneer opens the ballot box. He/she reveals each sealed price, determines winning price, the $(M+1)$st highest price, and finds the wining bidders who bid higher than the winning price. At the end, each winning bidder buys one unit of the goods at the $(M+1)$st winning price. If more than M bidders bid at the same highest price, the auction fails [3].

If $M = 1$, it is equivalent to the well-known Vickrey auction. Starting with the work by Nurmi & Salomaa [111] and Franklin & Reiter [64], a lot of secure sealed-bid auction schemes have been proposed, e.g., [74, 108, 126, 125, 10, 3, 24, 97, 84, 88, 26, 117, 41].

Vickrey auction method is also used on eBay[4] for single unit auctions in a modified form. That is, instead of the winning price being the highest losing bid, the highest losing bid plus a minimum bid increment is used [5, 79].

For more information about auctions and auction types, we refer to [147, 118, 99, 90, 55].

3.2 Differences Between Auction, Procurement and Tendering

This section is intended to clarify confusions that may arise when discussing the business processes of *auction*, *procurement* and *tendering*. They share some common procedural steps and properties; for example, bidder registration, bidding submission and, possibly, winner determination steps. However, Du says in his Ph.D. thesis [53] that there are many differences especially between auction and tendering which are also summarized in Table 3.2.

As seen from the Table 3.2, the roles of buyer and seller in an "auction" is reversed compared to those in a "procurement" and "tendering". In an "auction", many buyers make an offer to the seller; whereas in a "procurement" and "tendering" many sellers make an offer to the buyer. This is one of the basic differences. In "tendering", a buyer specifies what type of goods he/she wants to buy, and values each tender with the assessment of additional factors such as quality of service or

[3] *William Spencer Vickrey* (1914-1996) was a Canadian professor of economics who wrote the first game-theoretic analysis of auctions in 1961 (including the second-price auction [146]). He was awarded the Nobel Memorial Prize in Economics with James Mirrlees for their research into the economic theory of incentives under asymmetric information [110].

[4] *eBay Inc.* is a global commerce and payments leader, providing a robust system where buyers and sellers of all sizes can compete and win with 119.7 millions active users as of June 30, 2013 [56].

Properties	Auction	Procurement	Tendering
Bidding mechanism	English, Dutch, First-price sealed-bid, Vickrey	Reversed auctions	Mainly reversed sealed-bid
Products	Single items or lot	Projects, design, services, single item or lot	Construction project, government purchasing
Buyer	Many	One	One
Seller	One	Many	Many
Bidding party	Buyers	Sellers	Sellers
Winner resolution attribute	Single attribute price	Multi-attribute	Multi-attribute such as price and quality

Table 3.2: Differences Between Auction, Procurement and Tendering [53]

time of delivery. Thus, the winner in a "tendering" may not be the one who bids the lowest price. On the other hand, in an "auction" the winner is determined by using only the price factor, whether the auction is a traditional English, Dutch, first-price sealed-bid or Vickrey auction [53].

The word *auction* is used in many papers to refer to all contracting methods involving a bidding process. In this thesis, we follow the definition from Teich et al. [137] shown in Table 3.3. Based on Guttman and Maes's [73] online market analysis, Teich et al. [137] classifies the online market using the number of buyers and sellers involved in an online business negotiation. This definition also highlights the differences between "auction" and "tendering" [53].

		BUYERS	
		ONE	MANY
SELLERS	ONE	Negotiation	Auction
	MANY	Reverse Auction	Markets

Table 3.3: Market Framework [137]

One buyer and one seller defines the traditional business negotiation. One seller and many buyers define the auction type of negotiation. Many sellers and one buyer define a reverse auction (also called as supply auction), exemplified by government procurement with a bidding process. Many buyers and many sellers represent a market place. The "tendering" definition is procurement with a bidding process which falls into the reverse auction category [53].

3.3 General Security Issues of Electronic Auctions

In this section we discuss the most common threats and security requirements for electronic auctions. A summarized security requirement analysis was first done by Boyd and Mao [22] in 2000. They identified that internet auctions face potential threats for being abused in the following ways [53]:

- In *bid shielding*, the higher valued bid is withdrawn in the last minutes before bidding closes, allowing the lower price to be accepted as the winner. This is a problem directly associated with English Auction, which is an open, progressive ascending process.
- In *bid siphoning*, the seller monitors an auction and then makes a direct contact with a bidder and offers an equivalent item to the bidder. This way the seller can obtain a buyer for its goods without paying commission to the auction site. This will happen in a situation where bidders are not anonymous which applies to all types of auctions.
- *Shilling* is an abuse for driving bidding price up by inserting false bids in English open-cry auction. The seller colludes with the shill acting as a bidder in the crowd. If the shill wins the auction, the item will be moved to another auction for sale. It is even harder to detect shilling in an electronic auction. Shilling has application to Vickrey [146], where a false bid can be injected very closely to the first highest or lowest price.
- *Sniping* bidder enters his/her bid at the last moment hoping this will prevent other bidders from responding. It is associated with English auction.
- *Misrepresented or non-existent items* are the most common complaints of electronic (indeed online) auctions, because the bidder cannot physically examine the goods before bidding as in a traditional auction.

Fairness, confidentiality, anonymity and minimization of trust are the main and common desired security requirements [22].

- *Fairness* roughly means that all parties should be treated equally during the auction, e.g., the winner must pay for the goods, no bidder should get more information than other bidders or that the defined auction rules are followed.
- *Confidentiality* means that the losing bid should be kept secret even after opening. Peng et al. [119] state that confidentiality for sealed-bid auction means that no bid should be revealed before bid opening time. Keeping loser bidding secret is defined as another property which is called as *privacy* [119].
- *Anonymity* refers to bidders bid as an anonymous party. This generally excludes winning bidder's identity and so loser bidders' identity should be kept secret.

- *Minimization of trust* is trying to reduce the trust from the auctioneer.

These requirements varies in different types of auctions, therefore the definitions of these properties diversify from paper to paper. For instance, according to Sako [125], fairness, privacy and correctness are considered to be three major security issues in auction protocols. On the other hand, Peng et al. [119] divides sealed-bid auction properties into two: basic properties including correctness, confidentiality and fairness, and optional properties including anonymity, privacy and public verifiability as defined in the following.

- *Correctness* means that when every party runs the rule honestly, the correct winning price and winner(s) can be determined according to the auction rule.
- *Confidentiality* for sealed-bid auctions is that no bids can be published to any parties (including the auctioneer) until the bid opening step.
- *Fairness* can be defined by three concepts: no bidder knows anything about other bidders' bids before he/she submits his/her own bid; a submitted bid cannot be modified; no bidder should be able to deny his/her bid submission, which may be also called *non-repudiation* of bids.
- *Anonymity* means that the identities of losing bidders are kept secret.
- *Privacy* means that the losing bids remain confidential until the end of the auction even to the auctioneer.
- *Public verifiability* is publicly verifiable of the validity of the result of the auction.

For more information about security concepts for electronic auctions, we refer to [22, 88, 54, 53].

3.4 Some Selected Cryptographic Auction Protocols

In this section, we focus on roughly three selected auction protocols which are famous in the literature.

3.4.1 Auction Protocol of Naor et al.

The scheme by Naor et al. [108] is based on two servers (auctioneer and auction issuer) and bidders. The auction issuer need not to be a trusted third party but expected not to collude with the auctioneer. The protocol, which is a solution for a second-price sealed-bid auction, uses Yao's garbled circuits.

The protocol is roughly as follows (see Figure 3.2): Bidders first submit their encrypted bids to the auctioneer (the auction issuer can decrypt part of the encryption, but even it cannot reveal the actual bids). The auction issuer then generates a *garbled* Boolean circuit that computes the auction outcome for any given set of bids. The auctioneer forwards the portions of the bids to the auction issuer, which decrypts the bids in order to compute garbled inputs to the circuit. The auction issuer then sends the circuit and the inputs to the auctioneer, along with a signed translation table that decrypts the output of the circuit. The auctioneer uses garbled inputs and garbled circuit in order to evaluate and learn the garbled outputs. The auctioneer learns the actual result by using the signed translation table received from the auction issuer.

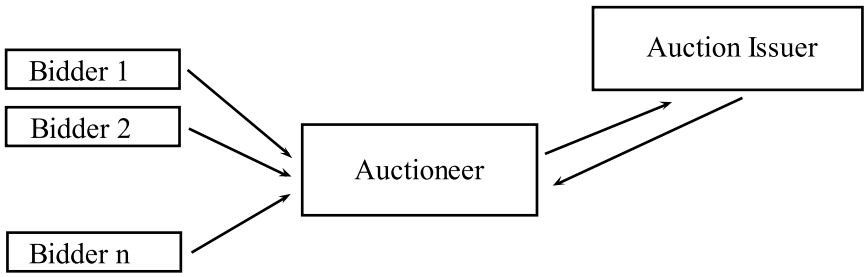

Figure 3.2: High-level Description of [108]

The protocol uses *pseudo-random functions* and *oblivious transfer*. The pseudo-random function $F_K(x)$ can be implemented by keying a block cipher with the key K and encrypting x, or keying a hash function with K and applying it to x. For the oblivious transfer, the scheme has a two-party protocol which is known as 1-out-of-2 oblivious transfer (1-out-of-2 OT). The OT protocol involves two parties, a *sender* that knows two secret values (m_0, m_1), and a *chooser* whose input is $\sigma \in \{0, 1\}$. At the end of the protocol, the chooser learns m_σ, while learning nothing about $m_{1-\sigma}$, and the sender learns nothing about σ.

Drawbacks of this system are the large communication complexity and detection of corrupted party is done only by using a cut-and-choose technique. Juels and Szydlo [84] removed a critical security flaw in the original protocol and based their version on RSA which results in less computational complexity for the bidders but even more complexity for the auction servers.

3.4.2 Auction Protocol of Lipmaa et al.

Lipmaa et al. [97] propose two cryptographic Vickrey auction schemes that work in the two-party model including bidders, a seller and an auction authority (auction issuer). The first scheme illustrates the basic properties and named as "simple auction scheme" in the paper [97]. According to authors this simple scheme has several vulnerabilities. The main contribution of the paper is "homomorphic auction scheme" which is the second scheme proposed in the paper.

The protocols are roughly as follows (see Figure 3.3): Bidders encrypt their bids

using the auction authority's public key and send them to the seller who checks first the signatures on them, then sorts the encrypted bids according to a prespecified method (e.g., in lexicographic ciphertext order), and publishes them. The auction authority then opens all the bids, sets the selling price (e.g., the second highest bid), sends it to the seller, and proves its correctness by applying a novel, sophisticated zero-knowledge proof. Winning bidders are required to claim that they won (violating non-repudiation).

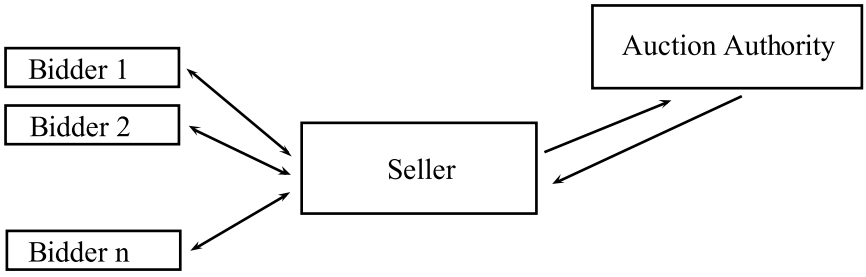

Figure 3.3: High-level Description of [97]

The protocol is very efficient, but only provides limited privacy as the selling price is published and the auction authority learns all bids. The only information hidden from the authority is the connection between bidders and bids. The number of possible bids is severely limited by saying "maximum allowed". Neither the seller nor the auction authority can manipulate the outcome without being detected. A collusion of both instances uncovers complete information, i.e., privacy can be invaded by seller and auction authority collusion.

3.4.3 Electronic Auction in Practice

In [20], Bogetoft et al., a group of two economists and nine computer scientists, present the implementation of a secure system for trading quantities of a certain commodity among many buyers and sellers, which also called as double auction. Particularly the deployed system was *used by Danish farmers* to trade contracts for *sugar beet* production on a nation-wide market. Since the system was developed using secure MPC, each bid submitted to the auction[5] is kept encrypted as from the submission time and no single party has access to the bids at any time. In addition, the system efficiently computes the price at which contracts are traded. This system, as far as is known, the *first large-scale practical application of secure MPC.*

The proposed protocol uses secure MPC technology as each bidder sends his/her bid in appropriately encrypted form to three parties. These parties are then compute the data while it is still in protected form. Thus, no single party ever has an access to any bid in clear form. However, by colluding, the parties can produce the desired output. The protocol also uses Shamir secret sharing scheme

[5] Sugar-beet double auction that took place in Denmark.

among three servers and a variant of a non-interactive verifiable secret sharing technique from [44].

The protocol is roughly as follows: The implemented auction protocol is an electronic double auction where the role of the auctioneer is played by three parties: the Danisco company, DKS (a sugar beet growers association) and SIMAP (Secure Information Management and Processing) project, which has been responsible for the practical application of multi-party computation, described in [20]. The auction mechanism is held in two phases (see Figure 3.4). In the first phase, each farmer logs into his/her existing account on Danisco website. This website is forwarded then to another web server and a Java applet is downloaded to the farmer's computer together with three public keys, each belonging to one of the auctioneers. Farmer then places his/her bid. The submitted bid is split into three pieces using secret sharing techniques. Each piece is then encrypted with a different public key and is sent back to the web server which stores them in a local database. In the second phase, the three parties send a representative who copies their shares from the web server database and uses their matching private keys to decrypt them. The market clearing price is then calculated using decrypted shares from the representatives and multi-party computation protocols.

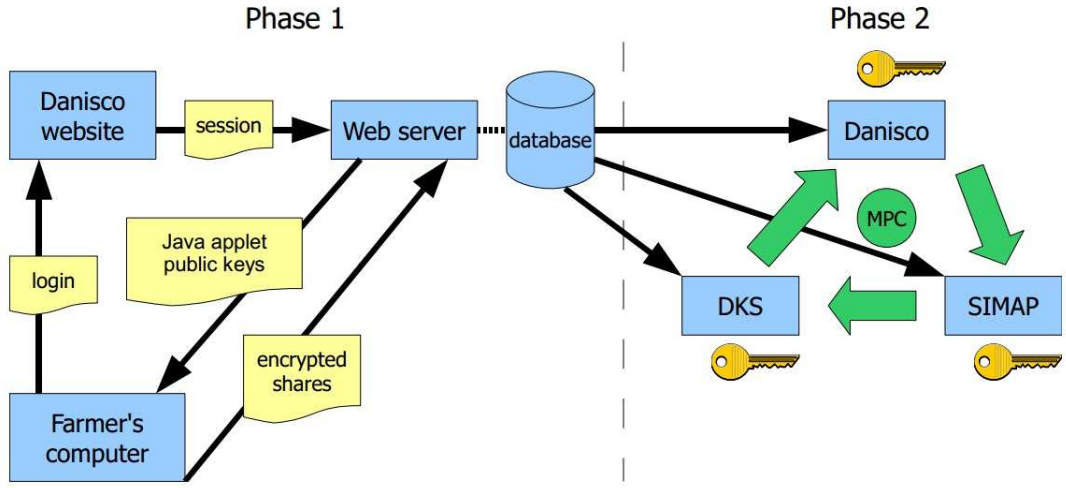

Figure 3.4: The Architecture Used in the Danisco Auction [18]

The authors selects a three party solution, partly because according to them it was natural in the given scenario, but also because it allowed using very efficient cryptographic protocols to do the secure computation.

3.5 Main Treasury Auctions in the World

In Treasury auctions, buyers (investors) typically submit their bids that specify an amount and a price (or a yield) at which they wish to purchase the amount demanded. Once submitted, these bids are sorted from the highest to the lowest

price (or from the lowest to the highest yield) and the amount for sale is awarded to the best bids (i.e., highest prices or lowest yields). In a *discriminatory-price auction*, the winning bidders pay the price they have bid (or the price according to the yield they have asked for), whereas in a *uniform-price auction*, they are all charged a price that is equal to the cut-off price, the highest market-clearing price [2].

In this section, we point out the Treasury auctions of United States, United Kingdom, Germany and Turkey. Later, we give more detailed information on Turkey case. For ease of understanding, we give an example for Turkish Treasury auctions.

3.5.1 The United States (US) Treasury Auctions

United States of America, in terms of public securities is known as the world's largest and most active market. Here is inspired by practices in many countries [93].

The modern auction process for bills, notes, and bonds begins with a public announcement by the US Treasury. Bids are accepted up to thirty days before the auction, and *submitted electronically through the Treasury Automated Auction Processing System (TAAPS)* and *by mail*. All bids are confidential and are kept sealed until the auction date. Many of the securities bought by large dealers will later be sold and resold on the secondary market to companies, banks, other dealers, and individuals. The primary dealers submit their competitive bids through TAAPS at the last possible moment, sometimes literally seconds before the deadline. Currently, the bids submitted through TAAPS are consolidated at the Federal Reserves in New York, Chicago, and San Francisco. Immediately after the auction deadline, these bids are reviewed and processed in these locations to assure compliance under the Treasury's Uniform Offering Circular. The bids at each of the review sites are sorted and then reviewed electronically by the US Treasury in Washington [61].

The US Treasury officials work their way down the list of bids, accepting the highest bid prices until all the securities have been awarded. All lower bids are rejected. In a $10 billion US Treasury bill auction example below (Table 3.4), securities would be awarded to the first four bidders only, whose bids total $12 billion. The two highest bidders would be awarded their total bid amounts, whereas the two bidders at the 5.10% discount rate would each receive $2 billion in securities [61].

The securities of the US Treasury are sold to the public through uniform-price auctions, i.e., the price of securities equals the highest accepted yield (coupons securities such as notes) or the highest accepted discount rate (bills). The list of accepted and rejected bids is not published but the total amount of bids offered and total amount of bids accepted are made available. Moreover, the high, low, and weighted averages of the price, discount rate, and equivalent bond yield of

NAME	BID PRICE	DISCOUNT / YIELD	AMOUNT
Bidder 1	$987.16	5.08% / 5.22%	$3.5 billion
Bidder 2	$987.13	5.09% / 5.23%	$2.5 billion
Bidder 3	$987.11	5.10% / 5.24%	$3.0 billion
Bidder 4	$987.11	5.10% / 5.24%	$3.0 billion
Bidder 5	$987.08	5.11% / 5.25%	$2.0 billion
Bidder 6	$987.06	5.12% / 5.26%	$1.0 billion

Table 3.4: Treasury Bill Auction Example in USA [61]

the accepted bids are published. Lastly, the final price, discount rate, and yield is also published within two hours of the auction [61].

On a report [144] issued in 2006, it is said that the Federal Reserve Banks did not apply strong encryption techniques to sensitive data and network traffic of such auctions. That is, weak encryption algorithms, such as the user's session information and application configuration files, were used as well as weak encryption format used to store and transmit certain passwords. These weaknesses allow an attacker to view data and use that knowledge to gain access to sensitive information including auction data.

3.5.2 The United Kingdom (UK) Treasury Auctions

A gilt is a UK Government security issued by Her Majesty's Treasury (HM Treasury). The name *Gilts* is short for *Gilt-edged stock*. The market has given this name to British Government securities because of their reputation as one of the safest investments [141].

The UK Government uses two different auction formats to issue gilts. The first one is the *conventional gilts* that are issued through a multiple price auction; and the second one is the *index-linked gilts* that are auctioned on a uniform price basis. The two different formats are applied because of the different nature of the risks involved to the bidder for the different securities [143]. *Conventional gilt auctions* are held on a bid price basis, i.e., successful bidders pay the price that they bid, with non-competitive bids allocated at the average accepted price. *Index-linked gilt auctions* are held on a single price basis, i.e., all successful bidders pay the lowest accepted price), with non-competitive bids also allocated at this lowest accepted price [142].

Except small retail bids from members of the DMO's Approved Group scheme, all bids at gilt auctions must be *submitted by, or through, a recognized primary dealer firm, via the Bloomberg Auction System (BAS)*. In emergency circumstances only, *a direct telephone line* to the DMO's dealing desk can be used. Thus, the investors wishing to participate in the auction process must submit their bid to a Gilt-edged Market Maker (GEMM) firm of their choosing, who is in turn obliged to submit that bid to the DMO, without charge. Dealers bidding on behalf of clients must

have a client code, which is allocated and maintained by the DMO dealers, to be used in the relevant field on their bid input screen. These codes can be taken by e-mail, Bloomberg message, company fax or letter and they are allocated on a once-only basis, and will be retained for use in future auctions [142].

Auctions of gilts are currently conducted as follows. The details of the gilt and the amount to be sold is announced in advance. Before the auction deadline the GEMMs submit their bids. The UK Debt Management Office's computer sorts these bids by price and the cut-off is found. Then all bids above the cut-off are filled in full; some fraction of each bid exactly at the cut-off is filled; all bids below the cut-off are rejected. The cut-off price, the proportion of bids at the cut-off that are accepted, the highest and average prices of the accepted bids, along with the quantity of bids received are published after the auction.

3.5.3 The Germany Treasury Auctions

The German Federal Government generally places single issues by the auctions. Only members of the "Auction Group Bund Issues" can participate directly in these auctions. The auctions are *carried out via the Deutsche Bundesbank Bund Bidding System (BBS)* [48]. BBS is an electronic primary market platform which is easily accessible and user-friendly while conforming security requirements. Users are authenticated to the Deutsche Bundesbank's ExtraNet by means of user IDs and passwords. If there is a problem while accessing to BBS web application via the internet, the bidders will be able to submit bids by fax as a backup solution. Bids can be submitted using optimized bidding masks. Just after submitting bids, bidders are informed whether their bids have been successfully accepted by the system. Bids can be viewed and deleted at any time until the end of bid submission phase [50]. The result of the auction is published on the day of the auction after the close of the bidding and directly after the allotment decision. They are published first in the BBS, to which the members of the Auction Group Bund Issues are linked. The data is then published shortly afterwards on the usual capital market information systems [49].

3.5.4 The Turkish Treasury Auctions

Before presenting the current Treasury auction process of Turkey, we need to define a few basic terms.

- *Auction date* is the day when the auction is held.
- *Value date* is the day when the interest for GDDSs begins to accrue. In many markets this is the same as the settlement date.
- *Settlement date* is the date by which an executed security trade must be settled. That is, it is the date by which a bidder must pay for the securities delivered by the Treasury.

- *Maturity date* is the date on which the issuer of a debt instrument must repay the principal in total. For example, a bond with a period of 5 years has a maturity date 5 years after its issue. The maturity date also indicates the period of time during which the lender or bondholder will receive interest payments.

- *Primary Dealership System* can be described as a system which is designed with the purpose of reducing roll-over risk, broadening investor base, constituting transparent, competitive and more organized market and also increasing liquidity and reducing volatility in the secondary market by giving certain official rights and obligations related to primary and secondary market of government debt securities to a group of professional intermediaries. In Turkey, the Primary Dealership System has been implemented since May 2000 excluding the May 2001 - September 2002 period when there had been a suspension due to the negative financial conditions.

  Primary dealership systems are widespread in the management of sovereign debt and borrowing. Of the 27 EU members, only five have not launched a primary dealer system: Latvia, Estonia, Malta, Cyprus and Germany. Estonia has no primary dealer system because the government does not issue domestic debt securities, whereas in Lithuania it has been in operation for several years now. The system is common in other countries as well, such as the United States, UK, Japan, Canada and Brazil.

- *Primary Dealer* is a bank which has been selected according to some pre-set criteria in order to increase effectiveness of the auctions for GDDSs and of the transactions of the secondary market for the said notes. According to the Law on Regulating Public Finance and Debt Management, only banks can be assigned as a Primary Dealer in Turkey.

  In our proposed model, the bidders are assumed to be only the primary dealers.

Rules of the Auction Process

We may summarize the system (see Figure 3.5) as follows.

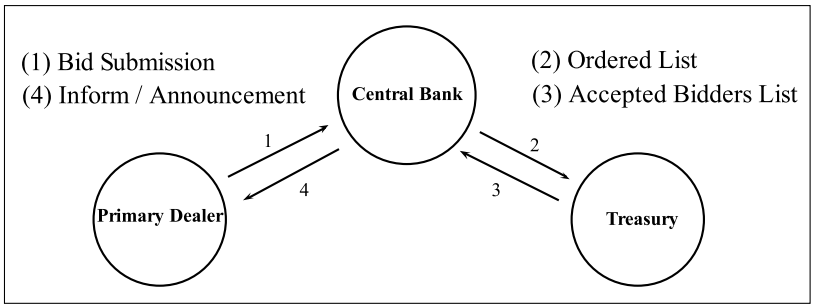

Figure 3.5: Current Treasury Auction System of Turkey

1. *Auction Announcement and General Issues*

   - Auction announcements are published on the Treasury website (see Table 3.5[6] for an example announcement) at least one day before the auction.
   - One year is calculated as 364 days in the Treasury auctions while it is 365 days on the secondary market.
   - The Central Bank runs all the operations related to auctions as the fiscal agent of the Treasury.

Auction Number	:	1754
Auction Date	:	15.01.2013
Settlement Date	:	16.01.2013
Maturity Date	:	09.04.2014
Maturity	:	15 Months (448 Days)
ISIN Code	:	TRT090414T19

Table 3.5: Treasury Auction Announcement Example in Turkey

2. *Auction Bids*

   - Retail and corporate investors can participate in the Treasury auctions through branches of the Central Bank, banks or brokers. While *banks can bid through EFT, brokers through TETS and insurance companies through fax*. Retail investors who will bid through the Central Bank branches need to present their citizen identification number. Since there is a great competition here, the largest part of the participation is provided by the primary dealers as corporate investors.
   - There is no limit on the number of investors.
   - Investors submit their bids (see Table 3.6 for an example bid information) in terms of price and nominal amount.

Expected yield	5,69	%
Unit price offered	94,617	
Nominal amount offered	1.000.000	TRY
Amount of payment to the Treasury on settlement date in case of being accepted	946.170	TRY
Amount of payment to the investor by the Treasury on maturity date	1.000.000	TRY

Table 3.6: An Example Bid Information of a Primary Dealer

   - All bids submitted are final bids for investors and investors are bound to their bids until the end of the auction.

[6] Taken from the Treasury Press Release dated 11.01.2013.

- If any of the bids submitted faulty by mistake, the investor should transmit his/her cancellation demand to the Treasury and the Central Bank via fax and phone, until the deadline of bidding for the auction. After bid submission phase, the bidders' cancellation demand is evaluated but the Treasury may or may not cancel the bid.
- There is no restriction for number of bids. So same investor may submit more than one bid. In the TRY[7] denominated auctions, retail and corporate investors can bid minimum 1,000 TRY and maximum 500 million TRY nominal in multiples of 1 TRY. Also, total bid amounts with the same price cannot exceed 500 million TRY.
- In USD denominated auctions, retail and corporate investors can bid minimum 10,000 USD and maximum 100 million USD and in multiples of 10,000 USD. Also, in EURO denominated auctions they can bid minimum 10,000 EURO and maximum 100 million EURO nominal in multiples of 10,000 EURO. Total bid amounts with the same price cannot exceed 100 million USD or EURO. Investors should deposit collateral of 1% of their total nominal bid amount before submitting the auction bids.
    - In TRY and FX[8] denominated auctions, collaterals for the rejected bids are returned to the investors in the auction day; whereas for the accepted bids, investors need to pay the remaining amount that is needed to be paid over the collateral in the value date.
    - In FX denominated auctions, retail and corporate investors who submit the bids via the Central Bank branches, the collateral is paid in effective amount of 1% of nominal bid amount or TRY that is calculated by the FX buying rates that the Central Bank announces on the auction day. If the investors' bids are rejected, the collateral is returned to the investor in FX or TRY on issue date. If the investors' bids are accepted and the collateral is paid in TRY, then the collateral is returned to the investor, after the required amount is fully paid by the investor. If the investors' bids are accepted and the collateral is paid in USD or EURO, then the investor pays the remaining amount on the value date.
- Public institutions are not obliged to pay collateral.
- A receipt reporting the bids submitted and a contact phone number are given to retail investors so that they can learn if their bids are accepted.
- The investors are not obliged to pay any stamp or seal duty.

3. *Announcements of Auction Results*

    - Auction results are published by the Central Bank. Related information is also announced on the Treasury website.

4. *Post-Auction Process*

[7] TRY: The currency abbreviation for the Turkish Lira.
[8] FX: Foreign Exchange

- If investors do not pay the required amount over the collateral, the collateral is recorded as revenue to the budget. These investors must attend at least 4 auctions with 20% of collateral. If these investors do not pay the required amount over the collateral, they must attend at least 4 auctions with 100% of collateral. After attending 4 auctions with the increased collaterals, investors may attend to the auctions with 1% collateral rate after the Treasury approval.
- Public institutions may pay the amount in which they win in TRY denominated auctions through non-competitive bids in USD or EURO. The amount will be calculated by the FX buying rates that the Central Bank announces on the value date.
- After auction process is completed, investors can buy securities in the secondary market through banks or brokers. At this stage, securities are subject to operations conducted between numerous buyers and sellers. The Treasury issues securities only to investors in the primary market.

5. *Redemption*

   - On the maturity date, payment is made through branches of the Central Bank or branches of Ziraat Bank which is the fiscal agent of the Central Bank.

Auction Evaluation and Award

After all the bids are submitted to the system, auction is closed and then all submitted bids are sorted from higher price to lower price (i.e., from lower interest to higher interest). After the preparation of the ordered list, it is then forwarded to the Treasury for the evaluation process. The Treasury examines all offers within the framework of existing conditions and determines the lowest price that is accepted. The offers whose prices are higher than that cut-off point are accepted and the others are rejected. In fact that cut-off point is the point where the required debt for the Treasury is also met. The following is the formula of finding cut-off point.

$$\sum_{i=1}^{m+1} \frac{p_i * a_i}{100} \geq a \quad \text{and} \quad \sum_{i=1}^{m} \frac{p_i * a_i}{100} < a$$

where p_i is the unit price and a_i is the nominal amount in i^{th} offer and a is the amount of required debt of the Treasury.

If there is an equality in offered unit prices of two primary dealers at the cut-off point, then the amount is shared as a weighted distribution in these two primary dealers. Sometimes, instead of this, the Treasury takes both of the primary dealers into the accepted ones and increments the cut-off point one point up. After the all operations and calculations (see below) are done, the results are submitted to the Central Bank in order to inform the bidders. Assuming there are k bids in an

Amount (Net, TRY Million)

	Offered	**Accepted**
Primary Dealers	2.534,0	380,1

Price (TRY)

	Offered	**Accepted**
Average Price	92,757	92,868
Minimum Price	92,560	92,850

Interest Rate (Average, %)

	Offered	**Accepted**
Term Rate	7,81	7,68
Annual Simple	6,34	6,24

Table 3.7: Treasury Auction Result Example in Turkey

ordered list, m is the cut-off point and d is the maturity in terms of days, then the following calculations are done by the Treasury corresponding to Table 3.7.

- Total Amount
  $$(\text{Offered, Accepted}) = (\mu_1, \mu_2) = \left(\sum_{i=1}^{k} \frac{p_i * a_i}{100} \ , \ \sum_{i=1}^{m} \frac{p_i * a_i}{100} \right)$$
- Total Nominal Amount
  $$(\text{Offered, Accepted}) = (\mu_3, \mu_4) = \left(\sum_{i=1}^{k} a_i \ , \ \sum_{i=1}^{m} a_i \right)$$
- Average Price
  $$(\text{Offered, Accepted}) = (\mu_5, \mu_6) = \left(\frac{\mu_1}{\mu_3} * 100 \ , \ \frac{\mu_2}{\mu_4} * 100 \right)$$
- Minimum Price
  $$(\text{Offered, Accepted}) = (p_k, p_m)$$
- Term Rate
  $$(\text{Offered, Accepted}) = (\mu_7, \mu_8) = \left(\frac{100 - \mu_5}{\mu_5} * 100 \ , \ \frac{100 - \mu_6}{\mu_6} * 100 \right)$$
- Annual Simple Rate
  $$(\text{Offered, Accepted}) = (\mu_9, \mu_{10}) = \left(\frac{364 * \mu_7}{d} \ , \ \frac{364 * \mu_8}{d} \right)$$

Auction results are announced to the public by the Central Bank with the title "Treasury Bills and Government Bonds Sold By Auctions". Also at the end, a press release is issued by the Treasury on its website (see Table 3.7[9]).

[9] Taken from the Treasury Press Release dated 11.01.2013.

After the value date, investors can buy government securities in the *secondary market* through banks or brokers. At this stage, government security is subject to operations conducted between numerous buyers and sellers. Note that the Treasury issues government securities only to investors in the primary market [121].

Example 3.2. Consider six valid bids are submitted for a Treasury auction with a maturity of 448 days as seen in Table 3.8. Each consists of four components: application number, name of the bank (bidder), unit price and nominal amount.

Order	**Name of the Bank**	**Unit Price (TRY 100)** p_i	**Nominal Amount** a_i
1.	Bank 1	94.80	30,000
2.	Bank 2	94.00	50,000
3.	Bank 3	94.50	50,000
4.	Bank 2	94.80	60,000
5.	Bank 4	95.00	30,000
6.	Bank 5	94.70	60,000

Table 3.8: Sample Bids Submitted by the Primary Dealers

After bid submission step, all the bids are sorted from higher unit price to lower unit price (i.e., from lower interest to higher interest) (see Table 3.9).

New Order	**Name of the Bank**	**Unit Price (TRY 100)** p_i	**Nominal Amount** a_i
~~5.~~ 1.	Bank 4	95.00	30,000
~~1.~~ 2.	Bank 1	94.80	30,000
~~4.~~ 3.	Bank 2	94.80	60,000
~~6.~~ 4.	Bank 5	94.70	60,000
~~3.~~ 5.	Bank 3	94.50	50,000
~~2.~~ 6.	Bank 2	94.00	50,000

Table 3.9: Ordered Sample Bids Submitted by the Primary Dealers

Then the new ordered list is sent from the Central Bank to the Treasury for evaluation. The Treasury examines all offers within the framework of existing conditions and determines the lowest price that is accepted. The offers whose prices are higher than that cut-off point are accepted and the others are rejected. In fact that cut-off point is the point where the required debt for the Treasury is also met. Let the required debt for the Treasury be $a = 175,000$ TRY. Then

using the formula

$$\sum_{i=1}^{m+1} \frac{p_i * a_i}{100} \geq a \quad \text{and} \quad \sum_{i=1}^{m} \frac{p_i * a_i}{100} < a$$

we get the cut-off point as $m = 3$ since

$$\sum_{i=1}^{4} \frac{p_i * a_i}{100} = 30,000 + 30,000 + 60,000 + 60,000 = 180,000 \geq 175,000$$

and

$$\sum_{i=1}^{3} \frac{p_i * a_i}{100} = 30,000 + 30,000 + 60,000 < 175,000.$$

Thus, first 3 bidders become the winners of this auction as shown in Table 3.10.

New Order	**Name of the Bank**	**Unit Price (TRY 100)** p_i	**Nominal Amount** a_i
1.	Bank 4	95.00	30,000
2.	Bank 1	94.80	30,000
3.	Bank 2	94.80	60,000
4.	Bank 5	94.70	60,000
5.	Bank 3	94.50	50,000
6.	Bank 2	94.00	50,000

Table 3.10: Cut-off Point of the Auction

After the determination of the winners, the following calculations are done. We have $k = 6$, $m = 3$ and $d = 448$ in this example.

- Total Amount

  $$\text{(Offered, Accepted)} = (\mu_1, \mu_2) = \left(\sum_{i=1}^{6} \frac{p_i * a_i}{100} \ , \ \sum_{i=1}^{3} \frac{p_i * a_i}{100} \right)$$

  $$= (264,890; 56,940)$$

- Total Nominal Amount

  $$\text{(Offered, Accepted)} = (\mu_3, \mu_4) = \left(\sum_{i=1}^{6} a_i \ , \ \sum_{i=1}^{3} a_i \right) = (280,000; 120,000)$$

- Average Price

  $$\text{(Offered, Accepted)} = (\mu_5, \mu_6) = \left(\frac{\mu_1}{\mu_3} * 100 \ , \ \frac{\mu_2}{\mu_4} * 100 \right) = (94.60; 94.90)$$

- Minimum Price
  (Offered, Accepted) $= (p_6, p_3) = (94.00; 94.80)$
- Term Rate
  (Offered, Accepted) $= (\mu_7, \mu_8) = \left(\frac{100 - \mu_5}{\mu_5} * 100 \ , \ \frac{100 - \mu_6}{\mu_6} * 100 \right)$
  $= (5.70; 5.37)$
- Annual Simple Rate
  (Offered, Accepted) $= (\mu_9, \mu_{10}) = \left(\frac{364 * \mu_7}{d} \ , \ \frac{364 * \mu_8}{d} \right) = (4.63; 4.37).$

Electronic auction versus domestic borrowing. We note that current auction schemes and protocols need significant modifications to be able to apply on the Treasury auctions. Some crucial different points can be described as follows:

- Consider a situation where one seller (the Treasury) and n bidders (the Primary Dealers) would like to make an agreement on the selling of GDDSs. Each bidder submits sealed-bids expressing how much he/she is willing to pay. The bidders want to be in the first m highest bidders in order to win the auction for a price that has to be determined by a publicly known rule (e.g., Section 3.5.4).
- Our proposed model does *not include trusted third party* and each bidder sends his/her encrypted bid to the server.
- In Treasury auctions, there are *more than one winner* and a *finding-cut-off-point step*, and the determination of winners depends on that cut-off point, that is, the number of the winners is not known until the end of the auction protocol. This number will depends on the amount of required debt of the Treasury and the nominal amount offers of the primary dealers.
- In Treasury auctions, *the name of the bidders* are also hidden whereas almost in all auction protocols the bidders are already known before running the protocol and during the protocol execution.

In this thesis, instead of modifying a generic auction protocol, we propose a new and secure protocol in Chapter 4 which is dedicated to only domestic borrowing.

CHAPTER 4

CONSTRUCTION OF A SECURE ELECTRONIC AUCTION MODEL

" Either exist as you are or be as you look."

— Mevlana

In Chapter 3, we discussed some international procedures about the Treasury auctions and described the Turkey case in details as a case study. In this chapter, we first introduce our proposed solution step-by-step and then prove its security. To the best of our knowledge, our proposed cryptographic protocol is the first solution to the Treasury auctions of Turkey.

4.1 Proposed Model

When the Treasury decides to hold an auction for issuing GDDSs, firstly it determines the amount of debt; secondly defines the auction time periods, i.e., open and close times, and announces these times to the public; thirdly informs the Central Bank in order to open the electronic bid submission system to the bidders. After then the Central Bank starts the system with the Treasury's confirmation. Thus, the bidders who want to participate the auction prepare their offers and use the system by submitting their bids within a certain time of period allowed. Starting with the bid submission, our proposed model consists of two phases:

- Submission and Evaluation phase,
- Award phase.

The Submission and Evaluation phase, as we said, starts with the bid submission of the bidders to the system where the bids are non-negative numbers. After bid submission deadline, secure function evaluation and secure MPC techniques are performed on those submitted (also encrypted) bids. This phase takes place between three parties:

- *The Primary Dealer*,
- *The Central Bank*,
- *The Treasury*.

In the Award phase, the auction results, i.e., the winners, are determined and announced. In this phase a protocol is run between the following two parties:

- *The Primary Dealer*,
- *The Treasury*.

Before presenting the details of the two phases of our proposed protocol, Table 4.1 can be glanced where the necessary notations are given.

Symbol	**Definition**
PD	Primary Dealer
CB	Central Bank
T	Treasury
a	Amount of required debt of the Treasury with $a \in \mathbb{Z}^+$
k	Number of bids in the auction with $k \in \mathbb{Z}^+$
i	Index where $i \in \{1, \ldots, k\}$
B_i	i^{th} bid participated in the auction for $i = 1, \ldots, k$
PD_i	Name of the Primary Dealer in B_i
p_i	Unit price offered in B_i and 6-bit-integer with $p_i \leq 100$
a_i	Nominal amount offered in B_i, $1,000 \leq a_i \leq 500,000$
y_i	Amount of payment calculated as $y_i = (p_i * a_i)/100$
pk_A	Paillier public key of party A
sk_A	Paillier secret key of party A
sk_A^j	j^{th} shared part of the secret key of party A
Sign_A	Process of time stamped RSA digital signing by party A
Ver	Process of verification of RSA digital signing
$\mathcal{E}_{pk}$	Paillier encryption under pk
$\mathcal{D}_{sk}$	Paillier decryption under sk

Table 4.1: Notations for the Proposed Model

The followings are the step-by-step protocol specifications. In the whole of our proposed model, it is assumed that there are k bids for each auction unless oth-

erwise noted. The encryption scheme used in the protocol is Paillier (see Section 2.2.2.2) and the signature scheme used is the RSA signature scheme (see Section 2.2.6.1). All the parties, i.e., the Primary Dealers, the Central Bank and the Treasury, have their own Paillier key pairs and Paillier secret key of each Primary Dealer is shared between that Primary Dealer and the Treasury. The parties also have their own RSA key pairs to be used in digital signature only. Here, we assume that the key distribution mechanism is secure.

The multiplicative inverse of x modulo n is denoted by x^{-1} and equals the integer y where $0 \leq y < n$, such that $x.y = 1 \mod n$. The multiplicative inverse is efficiently computed by using the Euclidean algorithm [100], and can also be used to negate an encrypted integer

$$\mathcal{E}_{pk}(-x) = \mathcal{E}_{pk}(x)^{-1} \mod n.$$

4.1.1 Submission and Evaluation Phase

The steps of this phase are as follows (see Figure 4.1 for the illustration).

1. Primary Dealer
   - Determines the unit price p_i and nominal amount a_i to be submitted.
   - Computes the amount of payment[1] $y_i = (p_i * a_i)/100$.
   - Forms the bid array $B_i := (PD_i, p_i, a_i)$.
   - Calculates the hash value of B_i and then signs it.
   - Sets $S_{B_i} := (\mathsf{Sign}_{PD_i}[\mathsf{Hash}(B_i)], \mathsf{Hash}(B_i))$.
   - Encrypts the values p_i, a_i and y_i using pk_T, and S_{B_i} using pk_{PD_i}.
   - Sets $X_i := (\mathcal{E}_{pk_{PD_i}}(S_{B_i}), \mathcal{E}_{pk_T}(p_i), \mathcal{E}_{pk_T}(a_i), \mathcal{E}_{pk_T}(y_i))$.
   - Sends[2] X_i to the system settled in the Central Bank.
2. Central Bank
   - Closes the system in order not to accept any new bids. Up to now, k four-tuple-bid values in the form

     $$X_i = (\mathcal{E}_{pk_{PD_i}}(S_{B_i}), \mathcal{E}_{pk_T}(p_i), \mathcal{E}_{pk_T}(a_i), \mathcal{E}_{pk_T}(y_i))$$

     are collected on the system.

[1] *Amount of payment* is the payment of the Primary Dealer will be paid to the Treasury on settlement date in case of being accepted in the auction. This means that the winner (or accepted) Primary Dealer pays p_i% of a_i TRY to the Treasury on settlement date and will take a_i TRY on maturity date.

[2] All the willing parties are required to send their own X_i's to the system within a certain time of period allowed, i.e., until the auction close time, through a **secure channel**, e.g., SSL connection.

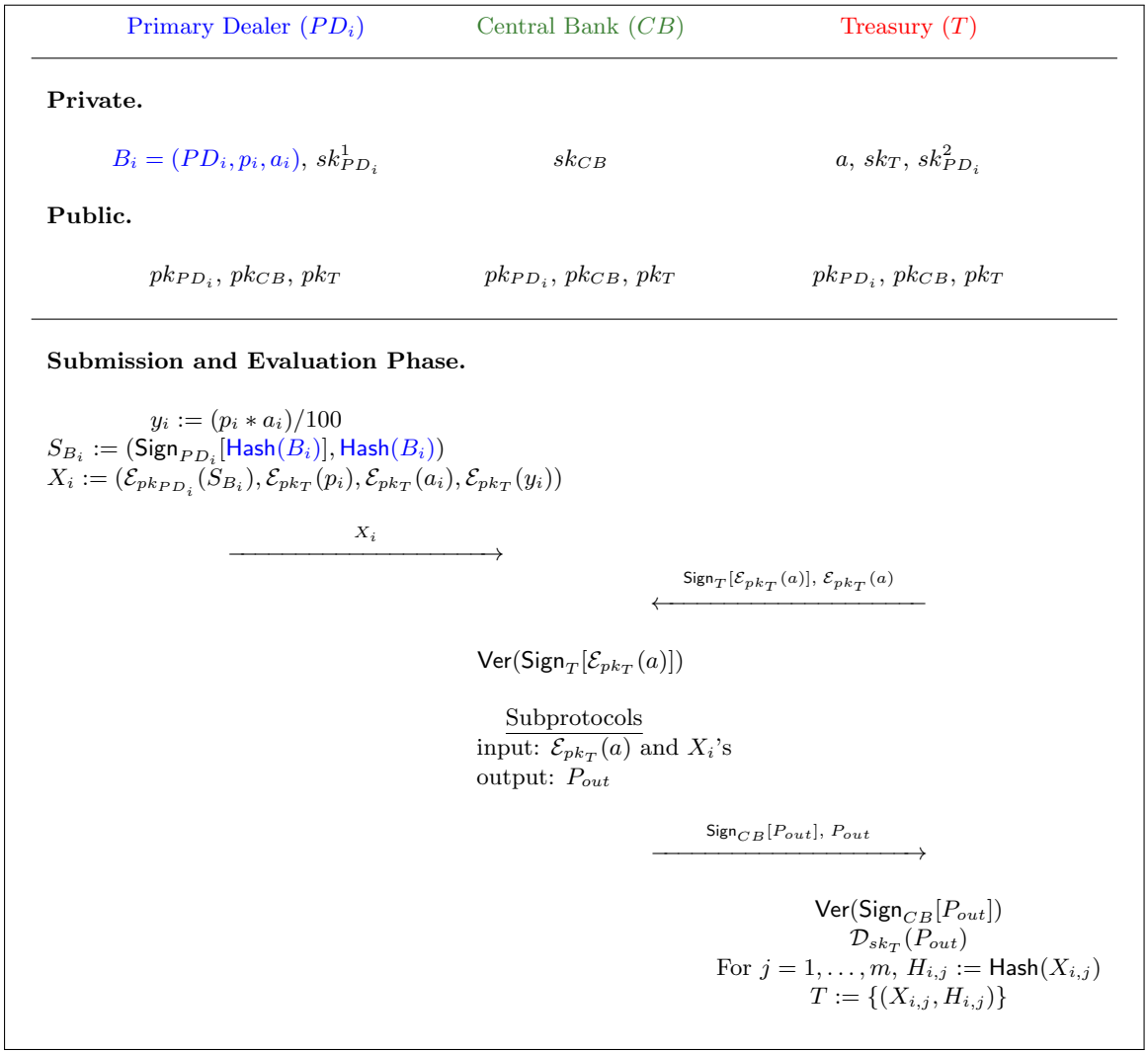

Figure 4.1: Submission and Evaluation Phase of Auction Process of GDDSs

3. Treasury

   - Encrypts the predetermined debt amount a which is secret to the Treasury and then signs[3] that encrypted value.
   - Sends $\mathsf{Sign}_T[\mathcal{E}_{pk_T}(a)]$ and the encrypted value $\mathcal{E}_{pk_T}(a)$ to the system settled in the Central Bank.

4. Central Bank

   - Verifies the signature on $\mathsf{Sign}_T[\mathcal{E}_{pk_T}(a)]$.

[3] While signing, the certificate of the signer is automatically appended to the signed value. So we consider that there is no need to mention about this again in next pages.

- Products all encrypted amount of payments, i.e., $\mathcal{E}_{pk_T}(y_i)$'s to get

$$\prod_{i=1}^{k} \mathcal{E}_{pk_T}(y_i) = \mathcal{E}_{pk_T}(\sum_{i=1}^{k} y_i) = \mathcal{E}_{pk_T}(\mu_1) = output_1.$$

- Products all encrypted nominal amounts, i.e., $\mathcal{E}_{pk_T}(a_i)$'s to get

$$\prod_{i=1}^{k} \mathcal{E}_{pk_T}(a_i) = \mathcal{E}_{pk_T}(\sum_{i=1}^{k} a_i) = \mathcal{E}_{pk_T}(\mu_3) = output_2.$$

- Runs **Subprotocols**.
  - The collected k-many X_i's considering as a list, are sorted in terms of unit prices ($\mathcal{E}_{pk_T}(p_i)$) by using the *insertion sorting*[4] method but in reverse order, i.e., sorting from largest to smallest instead of sorting from smallest to largest. The following is the pseudo-code of this algorithm for our proposed protocol.

Input: Array X with unordered elements [5]
Output: Array X with ordered elements [6]

function *insertionSort*(array X)
for $index = 1 \rightarrow k - 1$ **do**
  $temp = X[index]$
  $pre = index - 1$
  while $pre \geq 0$ and $Comparison(X[pre], temp)$ [7] **do**
    $X[pre + 1] \leftarrow X[pre]$
    $pre \leftarrow pre - 1$
  end while
  $X[pre + 1] = temp$
end for
return X

Comparison function (see Appendix A) is a two-party protocol proposed by Veugen [145]. Although there are other methods for private comparison, the reason for choosing the Veugen's method is that the party A holds two secret (encrypted) values $\mathcal{E}_{pk_B}(a)$ and $\mathcal{E}_{pk_B}(b)$ of ℓ-bits and the party B holds the private key. They wish to compare the numbers a and b. The actual values of a and b are not known to A and B. At the end, as we slightly changed, both party will learn the result, i.e., will learn the result whether $a \leq b$ or not without knowing a and b explicitly. Whereas, most of the

[4] *Insertion sorting* is a simple sorting algorithm that is relatively efficient for small lists and works by taking elements from the list one by one and inserting them in their correct position into a new ordered list [130].

[5] Unordered elements are X_i's with $1 \leq i \leq k$. In fact, here, the items to be sorted are the second elements of X_i's which are $\mathcal{E}_{pk_T}(p_i)$'s. Thus, while sorting $\mathcal{E}_{pk_T}(p_i)$'s, the other components of X_i move together with $\mathcal{E}_{pk_T}(p_i)$ while changing the places.

[6] Ordered new list is composed of $X_{i,j}$'s where i is the old place of X_i and j is the new place of X_i in the list. In this new list unit prices are sorted from largest to smallest.

[7] $Comparison(X[pre], temp)$ returns 1 if $\mathcal{D}_{sk_T}(X[pre]) \leq \mathcal{D}_{sk_T}(temp)$ and 0 if $\mathcal{D}_{sk_T}(X[pre]) > \mathcal{D}_{sk_T}(temp)$. See Appendix A for the details.

other methods have the property that there are two secret values and the party A holds one of them and the party B holds the other, and the parties want to compare those secret values (like Yao's Millionaire Problem [148]). At the end, both sides will learn whose value is greater than the other's. But in our case, despite we have two secret values, they are not held on two distinct parties, they both are held on one party, say on side A here. So our choice became the newly proposed method (2012) of Veugen [145] which is based on DGK comparison protocol [43].

– Changes the notation of X_i after sorting step as

$$X_{i,j} = (\mathcal{E}_{pk_{PD_{i,j}}}(S_{B_{i,j}}), \mathcal{E}_{pk_T}(p_{i,j}), \mathcal{E}_{pk_T}(a_{i,j}), \mathcal{E}_{pk_T}(y_{i,j}))$$

where j is the new place of X_i in the ordered new list.

– Takes the bid at the bottom, i.e., $X_{i,k}$ and on that tuple takes the encrypted unit price value[8],

$$\mathcal{E}_{pk_T}(p_{i,k}) = output_3.$$

– Uses all encrypted amount of payments in the ordered array, i.e., $\mathcal{E}_{pk_T}(y_i)$'s for $i = 1, \ldots, k$ as an input to the function *FindCutoffPoint* to find a positive integer m called *cut-off point* such that $m \leq k$ and

$$\sum_{i=1}^{m+1} \frac{p_i * a_i}{100} \geq a \quad \text{and} \quad \sum_{i=1}^{m} \frac{p_i * a_i}{100} < a.$$

The value m specifies the number of winners or the first m bids which are the accepted ones. Here is the algorithm.

Input: $\mathcal{E}_{pk_T}(a)$ and $\mathcal{E}_{pk_T}(y_i)$'s with $i = 1, \ldots, k$
Output: m where $m \leq k$

Function *FindCutoffPoint*$(\mathcal{E}_{pk_T}(a), \mathcal{E}_{pk_T}(y_1), \ldots, \mathcal{E}_{pk_T}(y_k))$
$m = 1$
for $t = 1$ to k **do**
  if *Comparison*$(\mathcal{E}_{pk_T}(a), \prod_{i=1}^{t} \mathcal{E}_{pk_T}(y_i))$ [9] **then**
    break for
  else
    $m = t$
  end if
end for
return m

[8] This value is the minimum price *offered*.

[9] *Comparison*$\mathcal{E}_{pk_T}(a), \prod_{i=1}^{t} \mathcal{E}_{pk_T}(y_i)$ returns 1 if $\mathcal{E}_{pk_T}(a) \leq \prod_{i=1}^{t} \mathcal{E}_{pk_T}(y_i)$ and 0 if $\mathcal{E}_{pk_T}(a) > \prod_{i=1}^{t} \mathcal{E}_{pk_T}(y_i)$. See Appendix A for the details.

Note that because of the additive homomorphic property of $\mathcal{E}_{pk_T}$ function, we have the equality $\prod_{i=1}^{t} \mathcal{E}_{pk_T}(y_i) = \mathcal{E}_{pk_T}(\sum_{i=1}^{t} y_i)$.

– Products the first m encrypted amount of payments in the list, i.e., $\mathcal{E}_{pk_T}(y_{i,j})$'s to get

$$\prod_{j=1}^{m} \mathcal{E}_{pk_T}(y_{i,j}) = \mathcal{E}_{pk_T}(\sum_{j=1}^{m} y_{i,j}) = \mathcal{E}_{pk_T}(\mu_2) = output_4.$$

– Products the first m encrypted nominal amounts in the list, i.e., $\mathcal{E}_{pk_T}(a_{i,j})$'s to get

$$\prod_{j=1}^{m} \mathcal{E}_{pk_T}(a_{i,j}) = \mathcal{E}_{pk_T}(\sum_{j=1}^{m} a_{i,j}) = \mathcal{E}_{pk_T}(\mu_4) = output_5.$$

– Takes the bid in m^{th} place of the list, i.e., $X_{i,m}$ and on that tuple takes the encrypted unit price value[10],

$$\mathcal{E}_{pk_T}(p_{i,m}) = output_6.$$

- Forms a set named P_{out} with $m + 6$ elements where

$$P_{out} := \{output_u : u = 1, \ldots, 6\} \cup \{X_{i,j} : j = 1, \ldots, m\}.$$

- Sends $\mathsf{Sign}_{CB}[P_{out}]$ and P_{out} to the Treasury.

5. Treasury

- Verifies the signature on $\mathsf{Sign}_{CB}[P_{out}]$.

- Decrypts the six encrypted values $\{output_u : u = 1, \ldots, 6\}$ and obtains the followings

$$\{\mu_1, \mu_2, \mu_3, \mu_4, p_{i,k}, p_{i,m}\}.$$

Here μ_1 is offered total amount of payment, μ_2 is accepted total amount of payment, μ_3 is offered total nominal amount, μ_4 is accepted total nominal amount, $p_{i,k}$ is offered minimum price and $p_{i,m}$ is accepted minimum price.

- Calculates average price offered $\mu_5 = \left(\frac{\mu_1}{\mu_3} * 100\right)$ and average price accepted $\mu_6 = \left(\frac{\mu_2}{\mu_4} * 100\right)$.

- Calculates term rate offered $\mu_7 = \left(\frac{100 - \mu_5}{\mu_5} * 100\right)$ and term rate accepted $\mu_8 = \left(\frac{100 - \mu_6}{\mu_6} * 100\right)$.

[10] This value is the minimum price *accepted*.

- Calculates annual simple rate offered $\mu_9 = \left(\frac{364 * \mu_7}{d}\right)$ and annual simple rate accepted $\mu_{10} = \left(\frac{364 * \mu_8}{d}\right)$ with d being the maturity in terms of days.
- Calculates the hash values of each accepted bids, i.e., $X_{i,j}$'s to get $H_{i,j} := \mathsf{Hash}(X_{i,j})$ for all $j = 1, \ldots, m$ and then prepares a look-up table T including m-many $(X_{i,j}, H_{i,j})$ couples.

4.1.2 Award Phase

In this phase, the Primary Dealers only learn the final decision on their corresponding submitted bids, i.e., they only learn whether the result is *Accept* or *Reject*. For this, we run the following two-party protocol steps (see Figure 4.2 for the illustration).

1. Primary Dealer
   - Calculates the hash value of X_i and sets $H_i := \mathsf{Hash}[X_i]$.
   - Sends H_i and his public key pk_{PD_i} to the Treasury.
2. Treasury
   - Checks if H_i matches one of $H_{i,j}$ values in the look-up table T.
   - Prepares the response res being either "Accept" or "Reject" and signs it as $\mathsf{Sign}_T[res]$. If H_i is equal to one of $H_{i,j}$'s then $res =$ "Accept", otherwise $res =$ "Reject".
   - Encrypts the signed response with the corresponding Primary Dealer's public key pk_{PD_i} as $\mathcal{E}_{pk_{PD_i}}(\mathsf{Sign}_T[res], res)$.
   - Decrypts that encrypted value with the second shared part of the secret key of the Primary Dealer, $sk^2_{PD_i}$ as $\mathcal{D}_{sk^2_{PD_i}}(\mathcal{E}_{pk_{PD_i}}(\mathsf{Sign}_T[res], res))$.
   - Sets $R := \mathcal{D}_{sk^2_{PD_i}}(\mathcal{E}_{pk_{PD_i}}(\mathsf{Sign}_T[res], res))$ and sends R to the Primary Dealer.
3. Primary Dealer
   - Decrypts R with the first shared part of the secret key of the Primary Dealer, $sk^1_{PD_i}$ and obtains the signed response.

$$\mathcal{D}_{sk^1_{PD_i}}\mathcal{D}_{sk^2_{PD_i}}(\mathcal{E}_{pk_{PD_i}}(\mathsf{Sign}_T[res], res)) = (\mathsf{Sign}_T[res], res).$$

   - Verifies the signature on $\mathsf{Sign}_T[res]$.

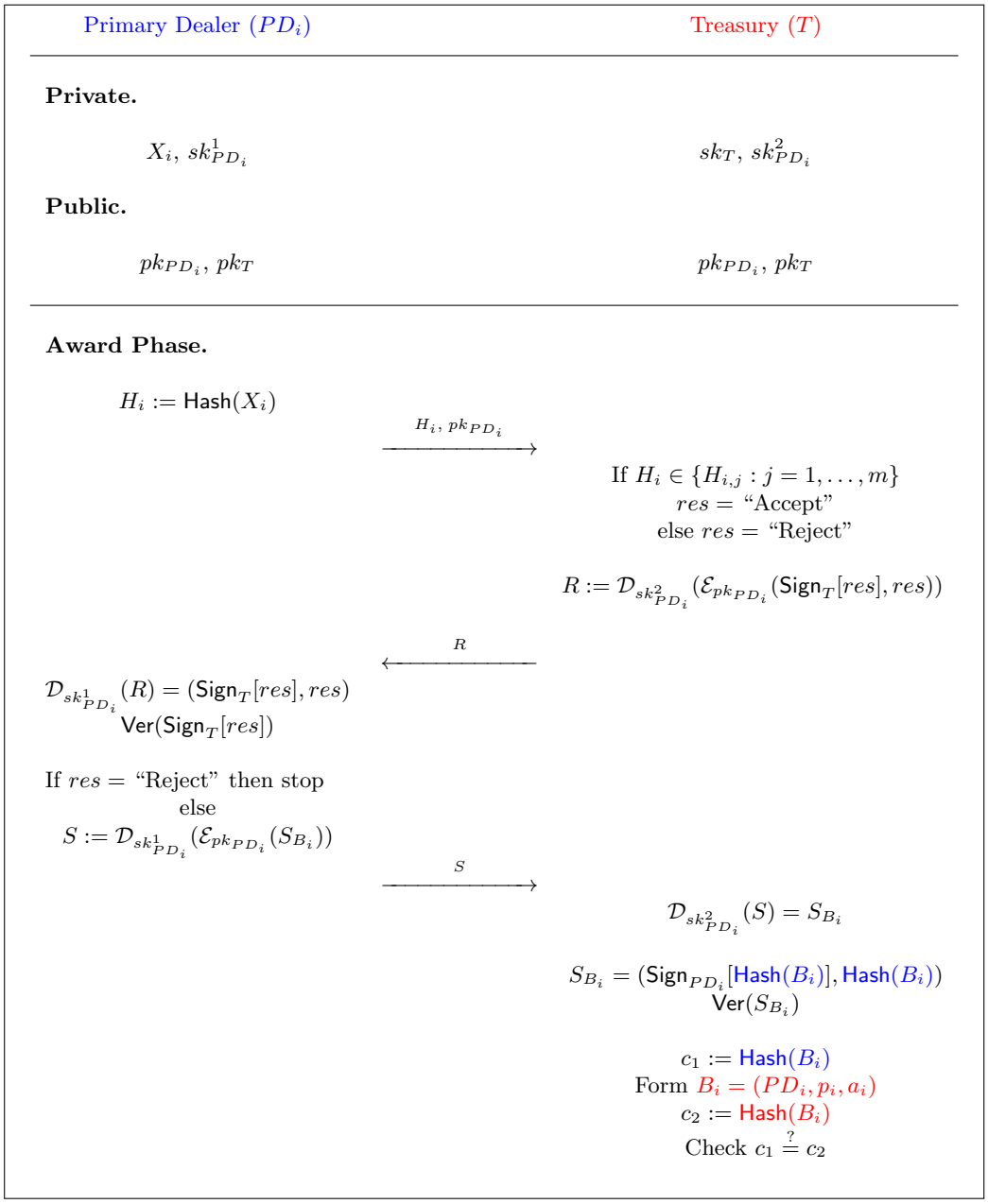

Figure 4.2: Award Phase of Auction Process of GDDSs

- If the response res = "Reject", does nothing and terminates the protocol. Otherwise, i.e., if the response res = "Accept" then continues with the next step.

- Gets the first component of the submitted bid X_i, i.e., $\mathcal{E}_{pk_{PD_i}}(S_{B_i})$ and decrypts it with the first shared part of the secret key of the Primary Dealer, $sk^1_{PD_i}$.

- Sets $S := \mathcal{D}_{sk^1_{PD_i}}(\mathcal{E}_{pk_{PD_i}}(S_{B_i}))$ and sends S to the Treasury.

4. Treasury

   - Decrypts S with its shared secret key $sk^2_{PD_i}$ and obtains S_{B_i}.
   - Verifies the signature on $S_{B_i} = (\mathsf{Sign}_{PD_i}[\mathsf{Hash}(B_i)], \mathsf{Hash}(B_i))$ and then sets $c_1 := \mathsf{Hash}(B_i)$.
   - Gets the corresponding Primary Dealer's name as PD_i and the corresponding unit price and nominal amount from look-up table to form the tuple $B_i = (PD_i, p_i, a_i)$. For this, decrypts $\mathcal{E}_{pk_T}(p_{i,j})$ with secret key sk_T and obtains $p_i := p_{i,j}$ where $j = 1, \ldots, m$ and decrypts $\mathcal{E}_{pk_T}(a_{i,j})$ with secret key sk_T and obtains $a_i := a_{i,j}$ where $j = 1, \ldots, m$.
   - Calculates the hash value of the formed tuple and sets $c_2 := \mathsf{Hash}(B_i)$.
   - Checks whether $c_1 \stackrel{?}{=} c_2$.
   - If confirmation occurs, then the formal contract is signed between the Treasury and each winner Primary Dealer. If confirmation does not occur, then there must be a problem with that Primary Dealer. If it is proven that there is an intended action then the penalty cases are concerned.

In this phase, if for example one winner remains silent, i.e., it does not start the protocol, one may consider various solutions, e.g., asking other users to prove that they really lose the auction. After detection of that silent winner, some penalty should be applied to it, e.g., banning from next few Treasury auctions, or imposing a fine. We underline that such a hiding player does not compromise privacy.

4.2 Security Analysis

In this section, we provide an informal security analysis of our proposed protocol in the presence of malicious parties from the Treasury, the Central Bank and the Primary Dealers. We assume a key distribution or establishment procedure has been successfully performed. That is to say, we focus on the adversaries against the auction protocol. Ensuring communication privacy, message integrity and reliable digital signing process are some crucial policies. We show that our overall protocol satisfies these policies and leaks no private information in the presence of malicious parties under the assumption that *the Treasury and the Central Bank do not collude*. Firstly, we note that malicious parties cannot see private inputs of honest primary dealers. This is because each input is encrypted using a randomized encryption scheme (e.g., Paillier) and the transmission is done through a secure channel. Secondly, the message integrity of all the values are satisfied by digital signature.

Attacks from external parties can be considered in practice, but such attacks are not special to our proposed system. Instead, we examine each party's malicious

case as in the following theorems.

Theorem 4.1. *A malicious Primary Dealer (bidder) cannot manipulate the outcome.*

Proof. For a primary dealer, the main privacy concern is secrecy of its name and anonymity of its bid values until end of the auction process. The name value PD_i is encrypted using a (2,2)-threshold encryption scheme and the names of the winners will not be decrypted until the winners are published. The name is revealed after the auction while the Treasury waits for the bidders to learn their own results.

Dishonest bidders cannot change the other party's inputs since all the bid components are encrypted and have signed parts. Finally, the response of i^{th} primary dealer can only be seen by that primary dealer itself because threshold decryption is performed by using the share $sk^1_{PD_i}$ which is known by only i^{th} primary dealer.

Note that the bidder may refuse to send the related hash value H_i to the Treasury at the beginning of the Award phase of the protocol. In this case both the bidder and the Treasury cannot learn the result of that bidder whether it is the accepted or rejected (because of anonymity of the bidders). In that case, the bidder must send the hash value H_i in order to finalize the overall outcome. We can prevent this type of problem by for example penalty cases (e.g., banning of participation for future auctions). In order to find out that malicious bidder, the Treasury and all the primary dealers meet and decrypt the related results who did not send its hash value. We underline that such a hiding bidder does not compromise privacy. □

Theorem 4.2. *A malicious Treasury obtains no information about bids except the winners.*

Proof. The Treasury obtains the ordered and the encrypted list of the accepted bidders and does not know any extra information about the bidders since the list is anonymised. For the Treasury, the only privacy concern is the secrecy of a. Since a is encrypted with pk_T, nobody else but only the Treasury itself can open (decrypt) this encrypted value and so also the Central Bank who makes some homomorphic evaluations with $\mathcal{E}_{pk_T}(a)$ can not learn any information about it. Therefore, a malicious Treasury cannot learn any additional information except the winners. □

Theorem 4.3. *A malicious Central Bank obtains no information about the bids.*

Proof. The Central Bank cannot see the sum values $\sum_{i=1}^{k} a_i$, $\sum_{i=1}^{k} y_i$ and also $\sum_{i=1}^{m} a_i$ and $\sum_{i=1}^{m} y_i$ clearly. Despite the Central Bank makes some evaluations and calculations with those values under encryption, it cannot extract the sum since it has no knowledge of the decryption key sk_T which is of the Treasury. Note that our proposed model does not consider active collusion between dishonest

parties in which secret keys are revealed. Hence, we may say that the privacy of the sums are also satisfied.

For the Central Bank only the privacy is of its secret key, sk_{CB}. The Central Bank only makes some evaluations, and uses his secret key only for signing the subprotocol outputs. Since the underlying subprotocols (sorting and comparing) are secure, a malicious Central Bank obtains no extra information. Hence, privacy will not be compromised in the presence of a malicious Central Bank. □

We do not consider the fairness in our proposal (which can be for example solved by gradual release bit commitment schemes [66]). However, even if either the Central Bank or the Treasury attempts to abort the protocol, this does not satisfy any advantage to any of the participant because all the bids are encrypted and signed. Hence, they cannot manipulate the result. Moreover, even if the Treasury aborts the protocol during the award phase this does not add any advantage since the encrypted bids remain anonymous. In practice, the Central Bank and the Treasury are two governmental bodies and they are the organizers of auctions, so their abortion of the protocol will affect the trusty of society. Therefore, it is better to focus on the abortion of the primary dealers. But, their abortion may realize only during the submission of their bids. Hence, this does not give any advantage to them except being out of the auction, i.e., no bids will be submitted to the Central Bank correctly.

4.3 Complexity Analysis

In this section, we present the *computational cost* of our proposed protocol. For simplicity, we will only count the expensive asymmetric operations since symmetric encryptions and hash functions can be ignored. Note that the submitted encrypted bid is a 4-tuple component. The Primary Dealers computes $4k$ encryptions where k denotes the number of bids. The Central Bank receives k four-tuple encrypted bids. After the bid submission deadline, subprotocol step will be run for k bids. We have $(k-1)k/2$ comparisons for k values in *Sorting* function and at most k comparisons for k values in *FindCutoffPoint* function. There are $(3\ell+10)$ public key encryptions in one *Comparison* function, then in total $(3\ell+10)(k^2+k)/2$ public key encryptions exist under the subprotocol step. Hence, in the Submission and Evaluation phase there are in total, with the Treasury's only one encryption, $(8k+2+(3\ell+10)(k^2+k))/2$ public key encryptions and 3 additional signatures. There are only one public key operation and one signature in the Award phase. Hence, there will be in total $(16k+24+(3\ell+10)(k^2+k))/2$ public key operations in our proposed model.

As for the *communication complexity*, there are in total $(4k+2\ell+4m+13)$ public key encryptions and 2 additional signatures transferred in the Submission and Evaluation phase, and one hashed value and 2 public key messages transferred in the Award phase. Hence, there will be in total $(4k+2\ell+4m+15)$ public key operations, 2 signatures and one hashed value transferred.

As for the *round complexity*, we have only constant rounds in our proposed protocol, i.e., 3 rounds for the Submission and Evaluation phase, 6 rounds for *Comparison* subprotocol and 3 rounds for the Award phase; in total $3 + 6 + 3 = 12$ rounds in our proposed model.

CHAPTER 5

CONCLUSION

" Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning."

— Albert Einstein

We conclude with summarizing the study and discussing the generalizations of our proposed model. In the preceding chapters, we first give some financial and cryptographic information including definitions and concepts that are used throughout the thesis, and then describe the current auction systems with some selected schemes and Treasury auctions of US, UK, Germany and Turkey for GDDSs by pointing out the security and privacy issues for the bids in the auctions.

In Treasury auctions, there are more than one winners, say m, where m is called *cut-off point*. While in determination step of m, the amount of required debt of the Treasury and the nominal amount offers of the primary dealers are needed. Without them, m cannot be calculated. Focusing on this crucial point which differs in some ways from other proposed auction models, we propose a new secure electronic auction model as current auction schemes and protocols need significant modifications to be able to apply on the Treasury auctions.

Our proposed protocol securely collects the bids and analyzes them for determining the winners in a GDDSs auction. Since the sensitive data of primary dealers (e.g., bid price and bid amount) is given to the system, the bids must be hidden until the end of the auction process. Except the winners, the rejected bidders' quotes are not disclosed. In our proposed model, we use the secure MPC where all the parties in the process do not have to trust each other and the sensitive data stays private throughout the process. In fact, this model, satisfying both confidentiality and privacy, is based on secure MPC, secret sharing and threshold homomorphic cryptosystem. To the best of our knowledge, our proposed secure electronic auction model is the first study applied on issuing domestic borrowing securities.

REFERENCES

[1] ASCII Code: The Extended ASCII Table, `http://www.ascii-code.com/`, retrieved July 15, 2013.

[2] K. Abbink, J. Brandts, and P. Pezanis-Christou, Auctions for Government Securities: A Laboratory Comparison of Uniform, Discriminatory and Spanish Designs, UFAE and IAE working papers, Unitat de Fonaments de l'Anàlisi Econòmica (UAB) and Institut d'Anàlisi Econòmica (CSIC), November 2002.

[3] M. Abe and K. Suzuki, M+1-st Price Auction Using Homomorphic Encryption, in *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems: Public Key Cryptography*, PKC'02, pp. 115–124, Springer-Verlag, London, UK, 2002, ISBN 3-540-43168-3.

[4] M. Alderson, K. Brown, and S. Lummer, Dutch auction rate preferred stock, Financial Management, (16), pp. 68–73.

[5] Auctus Development, Inc., Auction Types and Terms, `http://www.auctusdev.com/auctiontypes.html`, 2004, retrieved July 28, 2013.

[6] L. M. Ausubel, An efficient ascending-bid auction for multiple objects, American Economic Review, 94, 1997.

[7] L. S. Bagwell, Dutch Auction Repurchases: An Analysis of Shareholder Heterogeneity, Journal of Finance, 47(1), pp. 71–105, 1992.

[8] S. Bakkal and T. Gürdal, İç Borçlanmanın Türkiye Ekonomisi Üzerine Etkileri (The Effects of Domestic Borrowing on Turkey's Economy), Akademik İncelemeler, 2(2), pp. 147–173, 2007.

[9] P. S. L. M. Barreto and V. Rijmen, The Whirlpool Hashing Function, in *First open NESSIE Workshop*, Leuven, Belgium, 2000.

[10] O. Baudron and J. Stern, Non-interactive Private Auctions, in *Proceedings of the 5th Annual Conference on Financial Cryptography*, pp. 354–, 2001.

[11] M. Bellare and P. Rogaway, Optimal Asymmetric Encryption - How to Encrypt with RSA, pp. 92–111, Springer-Verlag, 1995.

[12] A. Ben-David, N. Nisan, and B. Pinkas, FairplayMP: a system for secure multi-party computation, in *Proceedings of the 15th ACM conference on Computer and communications security*, CCS'08, pp. 257–266, ACM, New York, NY, USA, 2008, ISBN 978-1-59593-810-7.

[13] M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest, A Fair Protocol for Signing Contracts (Extended Abstract), in *ICALP*, volume 194 of *Lecture Notes in Computer Science*, pp. 43–52, Springer, September 1985.

[14] M. Ben-Or, S. Goldwasser, and A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC'88, pp. 1–10, ACM, New York, NY, USA, 1988, ISBN 0-89791-264-0.

[15] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, The Keccak Reference, Submission to NIST (Round 3), `http://keccak.noekeon.org/Keccak-reference-3.0.pdf`, 2011, retrieved June 27, 2013.

[16] G. R. Blakley, Safeguarding Cryptographic Keys, in *Proceedings of the 1979 AFIPS National Computer Conference*, pp. 313–317, AFIPS Press, Monval, NJ, USA, 1979.

[17] D. Bogdanov, S. Laur, and J. Willemson, Sharemind: A Framework for Fast Privacy-Preserving Computations, IACR Cryptology ePrint Archive, 2008, p. 289, 2008.

[18] D. Bogdanov and R. Talviste, An improved method for privacy-preserving web-based data collection, Technical Report T-4-5, Cybernetica, `http://research.cyber.ee/`, June 2009, retrieved July 27, 2013.

[19] D. Bogdanov, R. Talviste, and J. Willemson, Deploying Secure Multi-Party Computation for Financial Data Analysis, in *Financial Cryptography*, pp. 57–64, 2012.

[20] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft, Financial Cryptography and Data Security, chapter Secure Multiparty Computation Goes Live, pp. 325–343, Springer-Verlag, Berlin, Heidelberg, 2009, ISBN 978-3-642-03548-7.

[21] C. Boutin, NIST Selects Winner of Secure Hash Algorithm (SHA-3) Competition, `http://www.nist.gov/itl/csd/sha-100212.cfm`, October 2012, retrieved August 1, 2013.

[22] C. Boyd and W. Mao, Security Issues for Electronic Auctions, Technical report, Hewlett-Packard Company, July 2000, `http://www.hpl.hp.com/techreports/2000/HPL-2000-90.pdf`, retrieved June 27, 2013.

[23] M. Brandly, History of Auctions, `http://mikebrandlyauctioneer.wordpress.com/auction-publications/history-of-auctions/`, 2013, auctioneer Blog, retrieved July 28, 2013.

[24] F. Brandt, A Verifiable, Bidder-resolved Auction Protocol, in *Proceedings of the 5th International Workshop on Deception, Fraud and Trust in Agent Societies*, pp. 18–25, 2002.

[25] F. Brandt, *Fundamental Aspects of Privacy and Deception in Electronic Auctions*, Ph.D. thesis, Technische Universität München, Garching, Germany, 2003.

[26] F. Brandt, How to Obtain Full Privacy in Auctions, International Journal of Information Security, 5(4), pp. 201–216, September 2006, ISSN 1615-5262.

[27] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, OpenPGP Message Format, RFC 2440 (Standards Track), November 1998.

[28] Central Bank of The Republic of Turkey, TCMB Kanunu'nda değişiklik yapılmasına dair 25.04.2001 tarihli ve 4651 sayılı Kanun ile getirilen yenilikler (the innovations introduced by the Law No:4651 dated 25.04.2001 ammending the Central Bank Law), `http://www.tcmb.gov.tr/yeni/banka/kanunacik.html`, 2001, retrieved June 27, 2013.

[29] D. Chaum, C. Crépeau, and I. Damgård, Multiparty unconditionally secure protocols, in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC'88, pp. 11–19, ACM, New York, NY, USA, 1988, ISBN 0-89791-264-0.

[30] D. Chaum, I. Damgård, and J. van de Graaf, Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result, in *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, CRYPTO'87, pp. 87–119, Springer-Verlag, London, UK, 1988, ISBN 3-540-18796-0.

[31] B. Cherowitzo, Math 5410 Modern Cryptology Lecture Notes on Digital Signature Schemes, `http://www-math.ucdenver.edu/~wcherowi/courses/m5410/ctcdss.html`, 2012, retrieved June 7, 2013.

[32] J. B. Clarkson, Dense Probabilistic Encryption, in *In Proceedings of the Workshop on Selected Areas of Cryptography*, pp. 120–128, 1994.

[33] M. Cooney, IBM Touts Encryption Innovation, `http://www.computerworld.com/s/article/9134823/IBM_touts_encryption_innovation`, June 2009, retrieved July 27, 2013.

[34] G. Coşkun, *Devlet Bütçesi: Türk Bütçe Sistemi (State Budget: Turkish Budget System)*, Turhan Kitabevi, Ankara, Turkey, 1986.

[35] R. Cramer and I. Damgård, *Multiparty Computation, an Introduction, in Contemporary Cryptology (Advanced Courses in Mathematics CRM Barcelona S.*, Birkhauser Verlag AG, 2005.

[36] R. Cramer, I. Damgård, and J. B. Nielsen, Lecture Notes on Multiparty Computation, an Introduction, May 2009.

[37] R. Cramer, I. Damgård, and J. B. Nielsen, Secure Multiparty Computation and Secret Sharing - An Information Theoretic Appoach (Book Draft), `http://www.daimi.au.dk/~ivan/mpc-book.pdf`, December 2012, retrieved August 2, 2013.

[38] R. Cramer and V. Shoup, A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack, pp. 13–25, Springer-Verlag, 1998.

[39] P. Cramton, E. Filiz-Ozbay, E. Y. Ozbay, and P. Sujarittanonta, Discrete Clock Auctions: An Experimental Study, Papers of peter cramton, University of Maryland, Department of Economics - Peter Cramton, 2010.

[40] V. P. Crawford and P.-S. Kuo, A dual dutch auction in taipei: the choice of numeraire and auction form in multi-object auctions with bundling, Journal of Economic Behavior & Organization, 51(4), pp. 427–442, 2003.

[41] B. Curtis, J. Pieprzyk, and J. Seruga, An Efficient eAuction Protocol, in *Proceedings of the The Second International Conference on Availability, Reliability and Security*, ARES'07, pp. 417–421, IEEE Computer Society, Washington, DC, USA, 2007, ISBN 0-7695-2775-2.

[42] J. Daemen and V. Rijmen, *The design of Rijndael: AES — the Advanced Encryption Standard*, Springer-Verlag, 2002, ISBN 3-540-42580-2.

[43] I. Damgård, M. Geisler, and M. Krøigaard, Homomorphic Encryption and Secure Comparison, International Journal of Applied Cryptography, 1(1), pp. 22–31, February 2008, ISSN 1753-0563.

[44] I. Damgård and R. Thorbek, Non-interactive Proofs for Integer Multiplication, in *Proceedings of the 26th annual international conference on Advances in Cryptology*, EUROCRYPT'07, pp. 412–429, Springer-Verlag, Berlin, Heidelberg, 2007, ISBN 978-3-540-72539-8.

[45] G. Demange, D. Gale, and M. Sotomayor, Multi-item auctions, Journal of Political Economy, 94(4), pp. 863–72, August 1986.

[46] T. Demos, Exactly What is a Dutch Auction?, `http://blogs.wsj.com/deals/2012/06/21/exactly-what-is-a-dutch-auction/`, June 2012, The Wall Street Journal, retrieved July 27, 2013.

[47] T. Derdiyok, 1980 Sonrası Borçlanma Politikaları (Borrowing Policies After 1980), Maliye Dergisi, (138), 2001.

[48] Deutsche Bundesbank, Auction Procedure, `http://www.deutsche-finanzagentur.de/en/institutional/primary-market/auction-procedure/`, retrieved July 17, 2013.

[49] Deutsche Bundesbank, Auction Results, `http://www.deutsche-finanzagentur.de/en/institutional/primary-market/auctions-results/`, retrieved July 17, 2013.

[50] Deutsche Bundesbank, BBS Bund Bidding System: Documentation Version 1.5, `http://www.bundesbank.de/Redaktion/EN/Downloads/Service/Service_fuer_Banken_und_Unternehmen/BBS/bbs_documentation.pdf?__blob=publicationFile`, retrieved July 30, 2013.

[51] W. Diffie and M. E. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, IT-22(6), pp. 644–654, November 1976, ISSN 0018-9448 (print), 1557-9654 (electronic).

[52] H. Dobbertin, A. Bosselaers, and B. Preneel, RIPEMD-160, a strengthened version of RIPEMD, Fast Software Encryption FSE'96, pp. 71–82, 1996.

[53] R. Du, *Secure Electronic Tendering*, Ph.D. thesis, Queensland University of Technology, Faculty of Information Technology, Brisbane, Australia, August 2007.

[54] R. Du, E. Foo, C. Boyd, and B. Fitzgerald, Defining Security Services for Electronic Tendering, in P. Montague and C. Steketee, editors, *Second Australasian Information Security Workshop (AISW2004)*, volume 32 of *CRPIT*, pp. 43–52, ACS, Dunedin, New Zealand, 2004.

[55] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, July 2010, ISBN 0521195330.

[56] eBay Inc., eBay Inc. Reports Strong Second Quarter 2013 Results, `http://www.ebayinc.com/in_the_news/story/ebay-inc-reports-strong-second-quarter-2013-results`, July 2013, retrieved July 28, 2013.

[57] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, Information Theory, IEEE Transactions on, 31(4), pp. 469–472, 1985, ISSN 0018-9448.

[58] U. Emek, Kamu İhalelerinde Rekabetin Sağlanması ve Korunması (Establishment and Maintenance of Competition in Public Auctions), Research Report DPT:2657, State Planning Organisation, Ankara, Turkey, 2002.

[59] S. Even, O. Goldreich, and A. Lempel, A Randomized Protocol for Signing Contracts, Commun. ACM, 28(6), pp. 637–647, June 1985, ISSN 0001-0782.

[60] F. J. Fabozzi and F. Modigliani, *Capital Markets: Institutions and Instruments*, Prentice Hall, 1996, ISBN 0133001873, 9780133001877.

[61] Federal Reserve Bank of New York, Treasury Auctions, `http://www.newyorkfed.org/aboutthefed/fedpoint/fed41.html`, April 2007, retrieved June 27, 2013.

[62] C. Fei, D. Kundur, and R. Kwong, Achieving Computational and Unconditional Security in Authentication Watermarking: Analysis, Insights, and Algorithms, in *Security, Steganography, and Watermarking of Multimedia Contents*, volume 5681 of *Proceedings of SPIE*, pp. 697–708, SPIE, December 2005.

[63] P.-A. Fouque, G. Poupard, and J. Stern, Sharing Decryption in the Context of Voting or Lotteries, in *Proceedings of the 4th International Conference on Financial Cryptography*, FC'00, pp. 90–104, Springer-Verlag, London, UK, 2001, ISBN 3-540-42700-7.

[64] M. K. Franklin and M. K. Reiter, The Design and Implementation of a Secure Auction Service, in *IEEE Symposium on Security and Privacy*, pp. 2–14, IEEE Computer Society, 1995, ISBN 0-8186-7015-0.

[65] J. Garay, B. Schoenmakers, and J. Villegas, Practical and Secure Solutions for Integer Comparison, in *Proceedings of the 10th International Conference on Practice and Theory in Public-Key Cryptography*, PKC'07, pp. 330–342, Springer-Verlag, Berlin, Heidelberg, 2007, ISBN 978-3-540-71676-1.

[66] J. A. Garay and M. Jakobsson, Timed release of standard digital signatures, in *Proceedings of the 6th international conference on Financial cryptography*, FC'02, pp. 168–182, Springer-Verlag, 2003.

[67] G. D. Gay, J. R. Kale, and T. H. Noe, (Dutch) Auction Share Repurchases, Economica, 63(249), pp. 57–80, 1996.

[68] C. Gentry, Fully Homomorphic Encryption Using Ideal Lattices, in *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC'09, pp. 169–178, ACM, New York, NY, USA, 2009, ISBN 978-1-60558-506-2.

[69] G. Giulioni and E. Bucciarelli, Agent's behaviour in a sequential dutch auction: evidence from the pescara wholesale fish market, Applied Economics Letters, 18(5), pp. 455–460, 2011.

[70] O. Göktürk, Multi-Unit Auctions: A Literature Review, Academic Study, Ministry of Economy, Ankara, Turkey, 2008, `http://www.ekonomi.gov.tr/upload/BF09AE98-D8D3-8566-4520B0D124E5614D/Osman_Gokturk.pdf`, retrieved July 27, 2013.

[71] O. Goldreich, S. Micali, and A. Wigderson, How to play ANY mental game or A Completeness Theorem for Protocols with Honest Majority, in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC'87, pp. 218–229, ACM, New York, NY, USA, 1987, ISBN 0-89791-221-7.

[72] F. Gul and E. Stacchetti, The english auction with differentiated commodities, J. Economic Theory, 92(1), pp. 66–95, 2000.

[73] R. H. Guttman and P. Maes, Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce, in *Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98)*, pp. 135–147, 1998.

[74] M. Harkavy, J. D. Tygar, and H. Kikuchi, Electronic Auctions with Private Bids, in *Proceedings of the 3rd conference on USENIX Workshop on Electronic Commerce - Volume 3*, WOEC'98, pp. 6–6, USENIX Association, Berkeley, CA, USA, 1998.

[75] W. Henecka, S. Kögl, A. R. Sadeghi, T. Schneider, and I. Wehrenberg, TASTY: Tool for Automating Secure Two-party Computations, in *Proceedings of the 17th ACM conference on Computer and communications*

security, CCS'10, pp. 451–462, ACM, New York, NY, USA, 2010, ISBN 978-1-4503-0245-6.

[76] K. Henry, *The Theory and Applications of Homomorphic Cryptography*, Master's thesis, University of Waterloo, Mathematics in Computer Science, Waterloo, Ontario, Canada, 2008.

[77] M. Hirt, U. M. Maurer, and V. Zikas, MPC vs. SFE : Unconditional and Computational Security, in *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pp. 1–18, Springer, December 2008.

[78] P. Hoffman and W. C. A. Wijngaards, Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC, RFC 6605 (Proposed Standard), April 2012.

[79] A. Hsiao, Understanding eBay Bidding, `http://ebay.about.com/od/gettingstarted/a/gs_bids_2.htm`, 2013, retrieved July 28, 2013.

[80] InvestingAnswers Financial Dictionary, Dutch Auction, `http://www.investinganswers.com/financial-dictionary/stock-market/dutch-auction-1277`, retrieved July 28, 2013.

[81] IPART (2007), Reforming Port Botany's links with inland transport, Technical report, New South Wales, Australia, 2007.

[82] Istanbul Stock Exchange, Sabit Getirili Menkul Kıymetler (Fixed Income Securities), `http://www.yatirimyapiyorum.gov.tr/media/9653/sabitgetirilimenkulkiymetler.pdf`, 2010, retrieved July 27, 2013.

[83] K. V. Jónsson, G. Kreitz, and M. Uddin, Secure Multi-Party Sorting and Applications, 2011.

[84] A. Juels and M. Szydlo, A Two-server, Sealed-bid Auction Protocol, in *Proceedings of the 6th International Conference on Financial Cryptography*, FC'02, pp. 72–86, Springer-Verlag, Berlin, Heidelberg, 2002, ISBN 3-540-00646-X.

[85] E. Katok and A. E. Roth, Auctions of Homogeneous Goods with Increasing Returns: Experimental Comparison of Alternative "Dutch" Auctions, Management Science, 50(8), pp. 1044–1063, 2004.

[86] R. S. Katti and C. Ababei, Secure Comparison Without Explicit XOR, CoRR, abs/1204.2854, 2012.

[87] A. S. Kelso and V. P. Crawford, Job matching, coalition formation and gross substitutes, Econometrica, 50, pp. 1483–1504, 1982.

[88] H. Kikuchi, (M+1)st-Price Auction Protocol, in *Proceedings of the 5th International Conference on Financial Cryptography*, FC'01, pp. 351–363, Springer-Verlag, London, UK, 2002, ISBN 3-540-44079-8.

[89] M. S. Kiraz, *Secure and Fair Two-Party Computation*, Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven, Nederland, August 2008.

[90] P. Klemperer, A Survey of Auction Theory, http://press.princeton.edu/chapters/s7728.pdf, 2004, retrieved August 2, 2013.

[91] D. Kohel, MATH 3024 Elementary Cryptography and Protocols Lecture Notes on Secret Sharing, http://echidna.maths.usyd.edu.au/kohel/tch/MATH3024/, 2004, retrieved July 27, 2013.

[92] V. Kolesnikov, *Secure Two-Party Computation and Communication*, Ph.D. thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 2006.

[93] C. Konya, İç Borçlanmada İhale Tekniklerinin Etkinliği - Türkiye İncelemesi (Effectiveness of Domestic Borrowing Tender Techniques - Study of Turkey), Expertise Thesis, Undersecretariat of Treasury, Ankara, Turkey, 2004.

[94] H. B. Leonard, Elicitation of Honest Preferences for the Assignment of Individuals to Positions, Journal of Political Economy, 91(3), pp. 461–79, June 1983.

[95] Z. Li and C.-C. Kuo, Revenue-maximizing dutch auctions with discrete bid levels, European Journal of Operational Research, 215(3), pp. 721–729, 2011.

[96] Z. Li and C.-C. Kuo, Design of Discrete Dutch Auctions with an Uncertain Number of Bidders, Annals of Operations Research, February 2013.

[97] H. Lipmaa, N. Asokan, and V. Niemi, Secure Vickrey Auctions without Threshold Trust, in *Proceedings of the 6th International Conference on Financial Cryptography*, FC'02, pp. 87–101, Springer-Verlag, Berlin, Heidelberg, 2003, ISBN 3-540-00646-X.

[98] A. Lysyanskaya, *Signature Schemes and Applications to Cryptographic Protocol Design*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States, September 2002.

[99] R. P. McAfee and J. McMillan, Auctions and Bidding, Journal of Economic Literature, 25(2), pp. 699–738, June 1987.

[100] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996, ISBN 0849385237.

[101] D. Micciancio, A first glimpse of cryptography's Holy Grail, Commun. ACM, 53(3), pp. 96–96, March 2010, ISSN 0001-0782.

[102] Microsoft, Data Confidentiality, http://msdn.microsoft.com/en-us/library/ff650720.aspx, December 2005, retrieved June 7, 2013.

[103] Microsoft, X.509 Technical Supplement, http://msdn.microsoft.com/en-us/library/ff647097.aspx, December 2005, retrieved June 7, 2013.

[104] Microsoft, Digital Signatures and Certificates, http://office.microsoft.com/en-us/word-help/digital-signatures-and-certificates-HA010354667.aspx, 2010, retrieved June 7, 2013.

[105] J.-F. Misarsky, How (Not) to Design RSA Signature Schemes, in *Proceedings of the First International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, PKC'98, pp. 14–28, Springer-Verlag, London, UK, 1998, ISBN 3-540-64693-0.

[106] M. Mortensen, *Secret Sharing and Secure Multi-party Computation*, M. Mortensen, 2007.

[107] A. Mukhamedov and M. Ryan, Improved Multi-party Contract Signing, in *Proceedings of the 11th International Conference on Financial Cryptography and 1st International Conference on Usable Security*, FC'07/USEC'07, pp. 179–191, Springer-Verlag, Berlin, Heidelberg, Germany, 2007, ISBN 3-540-77365-7, 978-3-540-77365-8.

[108] M. Naor, B. Pinkas, and R. Sumner, Privacy Preserving Auctions and Mechanism Design, in *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC'99, pp. 129–139, ACM, New York, USA, 1999, ISBN 1-58113-176-3.

[109] NIST Computer Security Division CSRC, Digital Signatures: Approved Algorithms, http://csrc.nist.gov/groups/ST/toolkit/digital_signatures.html, July 2013, retrieved August 2, 2013.

[110] Nobelprize.org, Press Release dated October 8, 1996, http://www.nobelprize.org/nobel_prizes/economic-sciences/laureates/1996/press.html, retrieved July 20, 2013.

[111] H. Nurmi and A. Salomaa, Cryptographic Protocols for Vickrey Auctions, Group Decision and Negotiation, 2, pp. 363–373, November 1993.

[112] B. Oğuz, Türkiye'de İç Borçlanma (Domestic Borrowing in Turkey), http://www.genbilim.com/content/view/6524/89/, 2009, retrieved May 15, 2013.

[113] Ş. Özbilen, *Türkiye'de Kamu İç Borçlanması ve Ekonomik Etkileri (Public Domestic Borrowing and Its Effects in Turkey)*, Atila Kitabevi, Ankara, Turkey, 1999, ISBN 9757285277, 9789757285274.

[114] P. Özdemir, *İç Borç Yönetimi ve Türkiye (Domestic Debt Management and Turkey)*, Master's thesis, Çukurova University, Institute of Social Sciences, Department of Economics, Adana, Turkey, 2009.

[115] İ. Öztürk, *İç Borçlanmanın Sermaye Piyasasına Etkileri (The Effects of Domestic Borrowing on Capital Market)*, İAB yayınları, İstanbul Altın Borsası, Ankara, Turkey, 2011, ISBN 9789759293246.

[116] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, EUROCRYPT'99, pp. 223–238, Springer-Verlag, Berlin, Heidelberg, 1999, ISBN 3-540-65889-0.

[117] D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. A. Thorpe, Practical Secrecy-Preserving, Verifiably Correct and Trustworthy Auctions, in *Proceedings of the 8th International Conference on Electronic Commerce: The New E-commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet*, ICEC'06, pp. 70–81, ACM, New York, USA, 2006, ISBN 1-59593-392-1.

[118] Paul Robert Milgrom and Robert James Weber, A theory of auctions and competitive bidding, Econometrica, 50(5), pp. 1089–1122, September 1982.

[119] K. Peng, C. Boyd, E. Dawson, and K. Viswanathan, Five Sealed-bid Auction Models, in *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21*, ACSW Frontiers'03, pp. 77–86, Australian Computer Society, Inc., Darlinghurst, Australia, 2003, ISBN 1-920682-00-7.

[120] D. Pointcheval, How to Encrypt Properly with RSA, RSA Laboratories' CryptoBytes, 5(1), pp. 9–19, 2002.

[121] Republic of Turkey Prime Ministry Undersecretariat of Treasury, Public Debt Management Report, Technical report, 2012.

[122] R. L. Rivest, The MD5 Message Digest Algorithm, 1992, RFC 1321.

[123] R. L. Rivest, A. Shamir, and L. M. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, 21(2), pp. 120–126, February 1978, ISSN 0001-0782 (print), 1557-7317 (electronic).

[124] S. Robicheaux and C. Herrington, Google's Dutch auction initial public offering, Journal of the International Academy for Case Studies, 13, pp. 7–11, 2007.

[125] K. Sako, An Auction Protocol Which Hides Bids of Losers, in *Public Key Cryptography, Third International Workshop on Practice and Theory in Public Key Cryptography, PKC 2000, Melbourne, Victoria, Australia, January 18-20, 2000, Proceedings*, volume 1751 of *PKC'00*, pp. 422–432, Springer-Verlag, London, UK, 2000, ISBN 3-540-66967-1.

[126] K. Sakurai and S. Miyazaki, A Bulletin-Board Based Digital Auction Scheme with Bidding Down Strategy -Towards Anonymous Electronic Bidding without Anonymous Channels nor Trusted Centers, 1999.

[127] B. Schoenmakers, Cryptography 2 (2WC13) / Cryptographic Protocols 1 (2WC17), Lecture Notes on Cryptographic Protocols, `http://www.win.tue.nl/~berry/2WC13/LectureNotes.pdf`, February 2013, retrieved July 27, 2013.

[128] B. Schoenmakers and P. Tuyls, Practical Two-Party Computation Based on the Conditional Gate, in P. J. Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pp. 119–136, Springer, 2004, ISBN 3-540-23975-8.

[129] SecureSCM, D9.1: Secure Computation Models and Frameworks, International University in Germany, Technische Universiteit Eindhoven, SAP AG, `http://pi1.informatik.uni-mannheim.de/filepool/publications/octavian_securescm/SecureSCM_D.9.1_v1.1.pdf`, July 2008, retrieved July 27, 2013.

[130] R. Sedgewick and K. Wayne, *Algorithms*, Addison-Wesley, 4th edition, 2011, ISBN 978-0-321-57351-3.

[131] A. Shamir, How to share a secret, Commun. ACM, 22(11), pp. 612–613, November 1979, ISSN 0001-0782.

[132] J. Shikata, Unconditional Security, `http://www.jsps.go.jp/j-bilat/fos_jf/data/jishi_02/abstract/05.pdf`, 2002, retrieved August 2, 2013.

[133] M. Shor, Dictionary of Game Theory Terms, Single Unit Auction, `http://www.gametheory.net/dictionary/Auctions/SingleUnitAuction.html`, January 2007, retrieved July 28, 2013.

[134] K. K. Sivaramakrishnan, SFWR 4C03 Computer Networks and Computer Security Lecture Notes on Symmetric Key Cryptosystem and Public Key Cryptography, `http://www4.ncsu.edu/~kksivara/sfwr4c03/lectures/lecture9.pdf`, March 2004, retrieved July 27, 2013.

[135] P. Sujarittanonta, *Design of Discrete Auction*, Ph.D. thesis, University of Maryland, College Park, Maryland, United States, 2010.

[136] L. Tavernini, MAT 3633 Numerical Analysis Lecture Notes on Lagrange Interpolation, `http://tavernini.com/arc/mat3633note01.pdf`, August 2011, retrieved July 27, 2013.

[137] J. Teich, H. Wallenius, and J. Wallenius, Multiple-Issue Auction and Market Algorithms for the World Wide Web, Decision Support Systems, 26(1), pp. 49–66, July 1999, ISSN 0167-9236.

[138] E. W. Tischhauser, *Mathematical Aspects of Symmetric-key Cryptography*, Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, May 2012.

[139] T. Toft, Sub-linear, Secure Comparison with Two Non-colluding Parties, in *Public Key Cryptography*, pp. 174–191, 2011.

[140] M. Tulloch, *Microsoft Encyclopedia of Security*, Prentice Hall, July 2003, ISBN 9780735618770.

[141] United Kingdom Debt Management Office, A Private Investor's Guide to Gilts, Fourth edition, `http://www.dmo.gov.uk/index.aspx?page=publications/Investor_Guide`, December 2004, retrieved July 27, 2013.

[142] United Kingdom Debt Management Office, Official operations in the gilt-edged market - an Operational Notice, http://www.dmo.gov.uk/docs/publications/operationalrules/Opnot20091120.pdf, November 2009, retrieved July 27, 2013.

[143] United Kingdom Debt Management Office, UK Government Securities: a Guide to 'Gilts', Tenth edition, http://www.dmo.gov.uk/index.aspx?page=publications/Investor_Guide, June 2012, retrieved July 27, 2013.

[144] United States Government Accountability, Information Security: Federal Reserve Needs to Address Treasury Auction Systems, http://www.gpo.gov/fdsys/pkg/GAOREPORTS-GAO-06-659/html/GAOREPORTS-GAO-06-659.htm, August 2006, retrieved August 2, 2013.

[145] T. Veugen, Improving the DGK Comparison Protocol, in *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*, WIFS'2012, pp. 49–54, Tenerife, Spain, December 2012.

[146] W. Vickrey, Counterspeculation, Auctions and Competitive Sealed Tenders, Journal of Finance, 16, pp. 8–37, March 1961.

[147] R. J. Weber, Multiple-Object Auctions, Discussion Papers 496, Northwestern University, Center for Mathematical Studies in Economics and Management Science, Evanston, Illinois, August 1981.

[148] A. C.-C. Yao, Protocols for Secure Computations, in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS'82, pp. 160–164, IEEE Computer Society, Washington, DC, USA, 1982.

[149] T. Ylonen and C. Lonvick, The Secure Shell (SSH) Authentication Protocol, RFC 4252 (Proposed Standard), January 2006.

[150] J. Yu, *Discrete Approximation of Continuous Allocation Mechanisms*, Ph.D. thesis, California Institute of Technology, Pasadena, California, United States, 1999.

[151] W. H. Yuen, W. S. Wong, Y. Chi, C. W. Sung, and W. S. Wong, Optimal Price Decremental Strategy for Dutch Auctions, in *Communications in Information and Systems*, pp. 411–434, 2002.

[152] Y. Zheng, J. Pieprzyk, and J. Seberry, HAVAL - A One-Way Hashing Algorithm with Variable Length of Output, in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, ASIACRYPT'92, pp. 83–104, Springer-Verlag, London, UK, 1993, ISBN 3-540-57220-1.

[153] P. Zimmermann, A. Johnston, and J. Callas, ZRTP: Media Path Key Agreement for Unicast Secure RTP, RFC 6189 (Informational), April 2011.

APPENDIX A

Comparison Function

Assume that a party A has two encrypted values $\mathcal{E}_{pk_B}(a)$ and $\mathcal{E}_{pk_B}(b)$ of ℓ-bits and the party B has the private key. They want to compare the numbers a and b whose actual values are not known to A and B. By the following function the party A **outputs**

$$Comparison(\mathcal{E}_{pk_B}(a), \mathcal{E}_{pk_B}(b)) = \begin{cases} \mathcal{E}_{pk_B}(1) & \text{if } a \leq b \\ \mathcal{E}_{pk_B}(0) & \text{if } a > b. \end{cases}$$

If the result is decrypted by party B then the **output** becomes

$$Comparison(\mathcal{E}_{pk_B}(a), \mathcal{E}_{pk_B}(b)) = \begin{cases} 1 & \text{if } a \leq b \\ 0 & \text{if } a > b. \end{cases}$$

This protocol is proposed in Veugen's paper [145]. Note that we can use other methods for secure comparison as well, e.g., [128, 65, 83, 139, 86]. In our proposed protocol, the encrypted unit prices are to be compared pair by pair and the parties are the Central Bank (party A) and the Treasury (party B). According to Veugen [145], the following protocol shows how to adjust the DGK comparison protocol with encrypted inputs such that perfect security is achieved towards B requiring only a small increase in computational and communication complexity. The difference with DGK comparison protocol [43] is the modified subprotocol with private inputs. See [145] for the details.

Let $0 \leq a, b < 2^{\ell} < n$ and n be the Paillier public key component used in the main protocol. The notation $(a \leq b)$ is used to denote the bit such that

$$(a \leq b) = \begin{cases} 1 & \text{if } a \leq b \\ 0 & \text{if } a > b \end{cases}$$

and $\oplus$ denotes the exclusive or of two bits.

Party A | Party B

input: $\mathcal{E}_{pk_B}(a)$, $\mathcal{E}_{pk_B}(b)$, sk_A, pk_A | **input:** sk_B, pk_B

Choose random r, $0 \leq r < n$
$\mathcal{E}_{pk_B}(z) := \mathcal{E}_{pk_B}(a).\mathcal{E}_{pk_B}(b)^{-1}.\mathcal{E}_{pk_B}(2^\ell + r) \mod n^2$

$\xrightarrow{\mathcal{E}_{pk_B}(z)}$

$\mathcal{D}_{sk_B}(\mathcal{E}_{pk_B}(z))$
$\alpha := r \mod 2^\ell$ | $\beta := z \mod 2^\ell$

modified DGK comparison *subprotocol* with private inputs:

input: α | input: β
output: δ_A | output: δ_B

Compute $\mathcal{E}_{pk_B}(d)$ where $d = (z < (n-1)/2)$
Compute $\mathcal{E}_{pk_B}(\beta_i)$ where $0 \leq i < \ell$

$\xleftarrow{\mathcal{E}_{pk_B}(d),\ \mathcal{E}_{pk_B}(\beta_i)}$

If $0 \leq r < (n-1)/2$ then set $\mathcal{E}_{pk_B}(d) = \mathcal{E}_{pk_B}(0)$
Compute for each i, $0 \leq i < \ell$
  if $\alpha_i = 0$ then $\mathcal{E}_{pk_B}(\alpha_i \oplus \beta_i) = \mathcal{E}_{pk_B}(\beta_i)$
  else $\mathcal{E}_{pk_B}(\alpha_i \oplus \beta_i) = \mathcal{E}_{pk_B}(1).\mathcal{E}_{pk_B}(\beta_i)^{-1} \mod n$
Compute $\tilde{\alpha} = (r - n) \mod 2^\ell$
Adjust $\mathcal{E}_{pk_B}(\alpha_i \oplus \beta_i)$ for each i, $0 \leq i < \ell$
  if $\alpha_i = \tilde{\alpha}_i$ then $\mathcal{E}_{pk_B}(w_i) = \mathcal{E}_{pk_B}(\alpha_i \oplus \beta_i)$
  else $\mathcal{E}_{pk_B}(w_i) = \mathcal{E}_{pk_B}(\alpha_i \oplus \beta_i).\mathcal{E}_{pk_B}(d)^{-1} \mod n$
Compute $\mathcal{E}_{pk_B}(w_i) := \mathcal{E}_{pk_B}(w_i)^{2^i} \mod n$ where $0 \leq i < \ell$
Choose a uniformly random bit δ_A
Compute $s = 1 - 2.\delta_A$
Compute for each i, $0 \leq i < \ell$
  $\mathcal{E}_{pk_B}(c_i) = \mathcal{E}_{pk_B}(s).\mathcal{E}_{pk_B}(\alpha_i).\mathcal{E}_{pk_B}(d)^{\tilde{\alpha}_i - \alpha_i}.\mathcal{E}_{pk_B}(\beta_i)^{-1}.(\prod_{j=i+1}^{\ell-1} \mathcal{E}_{pk_B}(w_j))^3 \mod n$
For random r_i of $2t$ bits; set $\mathcal{E}_{pk_B}(c_i) := \mathcal{E}_{pk_B}(c_i)^{r_i} \mod n$ for all i

$\xrightarrow{\mathcal{E}_{pk_B}(c_i) \text{ in random order}}$

Checks whether one of $\mathcal{D}_{sk_B}(\mathcal{E}_{pk_B}(c_i)) = 0$
if there is, set $\delta_B = 1$
else set $\delta_B = 0$
Compute $\mathcal{E}_{pk_B}(z \div 2^\ell)$

$\xleftarrow{\mathcal{E}_{pk_B}(z \div 2^\ell),\ \mathcal{E}_{pk_B}(\delta_B)}$

Compute $\mathcal{E}_{pk_B}((\beta < \alpha))$ as follows:
  if $\delta_A = 1$, set $\mathcal{E}_{pk_B}((\beta < \alpha)) := \mathcal{E}_{pk_B}(\delta_B)$
  else $\mathcal{E}_{pk_B}((\beta < \alpha)) := \mathcal{E}_{pk_B}(1).\mathcal{E}_{pk_B}(\delta_B)^{-1} \mod n^2$

Compute $\mathcal{E}_{pk_B}(\gamma) = \mathcal{E}_{pk_B}(z \div 2^\ell).\mathcal{E}_{pk_B}(r \div 2^\ell).(\mathcal{E}_{pk_B}((\beta < \alpha)))^{-1} \mod n^2$

$\xrightarrow{\mathcal{E}_{pk_B}(\gamma)}$

$\mathcal{D}_{sk_B}(\mathcal{E}_{pk_B}(\gamma)) = \gamma$
Compute $\mathcal{E}_{pk_A}(\gamma)$

$\xleftarrow{\mathcal{E}_{pk_A}(\gamma)}$

$\mathcal{D}_{sk_A}(\mathcal{E}_{pk_A}(\gamma)) = \gamma$

output: γ where $(\gamma = 1) \equiv (a \leq b)$

Example A.1. Let us give an example for ease of understanding the *Comparison* protocol. For this let

- $a = 3$ and $b = 5$ be two ℓ-bit-number to be compared where $\ell = 3$.
- $p = 7$ and $q = 13$.
- $n = p.q = 7.13 = 91$ and $n^2 = 91^2 = 8281$.
- $p-1 = 7-1 = 6$ and $q-1 = 13-1 = 12$; we can find t-bit-primes p_1 and q_1 such that p_1 divides $p-1$ and q_1 divides $q-1$. Choose $p_1 = q_1 = 3 = (11)_2$, then $t = 2$.

Here is the protocol outlined step-by-step.

Input: $\mathcal{E}_{pk_B}(3)$ and $\mathcal{E}_{pk_B}(5)$.

Expected output: 1 which means that $a \leq b$.

1. A chooses a $2t$-bit random $r = 9 = (1001)_2 < n = 91$.
2. $\mathcal{E}_{pk_B}(z) := \mathcal{E}_{pk_B}(3).\mathcal{E}_{pk_B}(5)^{-1}.\mathcal{E}_{pk_B}(2^3+11) = \mathcal{E}_{pk_B}(3-5+2^3+11) \mod 8281$ meaning that $z = 3 - 5 + 2^3 + 9 = 15$.
3. A sends $\mathcal{E}_{pk_B}(z)$ to B.
4. B decrypts $\mathcal{E}_{pk_B}(z)$ and gets $z = 15$.
5. B calculates $\beta := z \mod 2^\ell = 15 \mod 2^3 = 15 \mod 8 = 7$, $\beta = 7$.
6. A calculates $\alpha := r \mod 2^\ell = 9 \mod 2^3 = 9 \mod 8 = 1$, $\alpha = 1$.
7. Then modified DGK comparison *subprotocol* with private inputs α and β starts:
   $\alpha = 1 = (\alpha_2\alpha_1\alpha_0)_2 = (001)_2$
   $\beta = 7 = (\beta_2\beta_1\beta_0)_2 = (111)_2$
   (a) B calculates $d = (z < (n-1)/2) = (15 < (91-1)/2) = (15 < 45) \equiv 1$ where d is the bit informing whether a carry-over occurred.
   (b) B sends $\mathcal{E}_{pk_B}(d) = \mathcal{E}_{pk_B}(1)$ to A.
   (c) B sends $\mathcal{E}_{pk_B}(\beta_i)$'s, $0 \leq i < \ell$, which are $\mathcal{E}_{pk_B}(1)$, $\mathcal{E}_{pk_B}(1)$, $\mathcal{E}_{pk_B}(1)$ to A.
   (d) A checks whether $0 \leq r < (n-1)/2$ or not. Since it is so, A corrects $\mathcal{E}_{pk_B}(d)$ by setting $\mathcal{E}_{pk_B}(d) = \mathcal{E}_{pk_B}(0)$ meaning that $d = 0$.
   (e) A computes for each i, $0 \leq i < \ell$,
      - $1 = \alpha_0 \neq 0 \longrightarrow \mathcal{E}_{pk_B}(\alpha_0 \oplus \beta_0) = \mathcal{E}_{pk_B}(1).\mathcal{E}_{pk_B}(\beta_0)^{-1} \mod n$ meaning that $\alpha_0 \oplus \beta_0 = 1 - \beta_0 = 1 - 1 = 0$.
      - $\alpha_1 = 0 \longrightarrow \mathcal{E}_{pk_B}(\alpha_1 \oplus \beta_1) = \mathcal{E}_{pk_B}(\beta_1)$ meaning that $\alpha_1 \oplus \beta_1 = \beta_1 = 1$.

- $\alpha_2 = 0 \longrightarrow \mathcal{E}_{pk_B}(\alpha_2 \oplus \beta_2) = \mathcal{E}_{pk_B}(\beta_2)$ meaning that $\alpha_2 \oplus \beta_2 = \beta_2 = 1$.

(f) A computes $\tilde{\alpha} = (r - n) \mod 2^\ell = (9 - 91) \mod 2^3 = 6 \mod 8$ and so $\tilde{\alpha} = 6 = (\tilde{\alpha}_2\tilde{\alpha}_1\tilde{\alpha}_0)_2 = (110)_2$ where $\tilde{\alpha}$ is the corrected value of α in case a carry-over actually did occur.

(g) A adjusts $\mathcal{E}_{pk_B}(\alpha_i \oplus \beta_i)$ for each i, $0 \leq i < \ell$,

- $1 = \alpha_0 \neq \tilde{\alpha}_0 = 0 \longrightarrow \mathcal{E}_{pk_B}(w_0) = \mathcal{E}_{pk_B}(\alpha_0 \oplus \beta_0).\mathcal{E}_{pk_B}(d)^{-1} \mod n$ meaning that $w_0 = (\alpha_0 \oplus \beta_0) - d = 0 - 0 = 0$.
- $0 = \alpha_1 \neq \tilde{\alpha}_1 = 1 \longrightarrow \mathcal{E}_{pk_B}(w_1) = \mathcal{E}_{pk_B}(\alpha_1 \oplus \beta_1).\mathcal{E}_{pk_B}(d)^{-1} \mod n$ meaning that $w_1 = (\alpha_1 \oplus \beta_1) - d = 1 - 0 = 1$.
- $0 = \alpha_2 \neq \tilde{\alpha}_2 = 1 \longrightarrow \mathcal{E}_{pk_B}(w_2) = \mathcal{E}_{pk_B}(\alpha_2 \oplus \beta_2).\mathcal{E}_{pk_B}(d)^{-1} \mod n$ meaning that $w_2 = (\alpha_2 \oplus \beta_2) - d = 1 - 0 = 1$.

(h) We have $(w_2w_1w_0)_2 = 2^2.w_2 + 2^1.w_1 + 2^0.w_0$. Then, A computes $\mathcal{E}_{pk_B}(w_i) := \mathcal{E}_{pk_B}(w_i)^{2^i} \mod n$ where $0 \leq i < \ell$,

- For $i = 0$, $\mathcal{E}_{pk_B}(w_0) = \mathcal{E}_{pk_B}(w_0)^{2^0} \mod n$ meaning that $w_0 = (\mathbf{0})_2$.
- For $i = 1$, $\mathcal{E}_{pk_B}(w_1) = \mathcal{E}_{pk_B}(w_1)^{2^1} \mod n$ meaning that $w_1 = (\mathbf{1}0)_2$.
- For $i = 2$, $\mathcal{E}_{pk_B}(w_2) = \mathcal{E}_{pk_B}(w_2)^{2^2} \mod n$ meaning that $w_2 = (\mathbf{1}00)_2$.

(i) A chooses a uniformly random bit, say $\delta_A = 1$.

(j) A computes $s = 1 - 2.\delta_A = 1 - 2.1 = -1$.

(k) A computes

$$\mathcal{E}_{pk_B}(c_i) = \mathcal{E}_{pk_B}(s).\mathcal{E}_{pk_B}(\alpha_i).\mathcal{E}_{pk_B}(d)^{\tilde{\alpha}_i - \alpha_i}.\mathcal{E}_{pk_B}(\beta_i)^{-1}.(\prod_{j=i+1}^{\ell-1} \mathcal{E}_{pk_B}(w_j))^3 \mod n$$

for each i, $0 \leq i < \ell$ and for each j, $i < j < \ell$,

- $\mathcal{E}_{pk_B}(c_0) = \mathcal{E}_{pk_B}(-1).\mathcal{E}_{pk_B}(1).\mathcal{E}_{pk_B}(0)^{0-1}.\mathcal{E}_{pk_B}(1)^{-1}.(\mathcal{E}_{pk_B}(w_1).\mathcal{E}_{pk_B}(w_2))^3 \mod n$ meaning that $c_0 = -1 + 1 - 0 - 1 + 3(1 + 1) = 5$.
- $\mathcal{E}_{pk_B}(c_1) = \mathcal{E}_{pk_B}(-1).\mathcal{E}_{pk_B}(0).\mathcal{E}_{pk_B}(0)^{1-0}.\mathcal{E}_{pk_B}(1)^{-1}.\mathcal{E}_{pk_B}(w_2)^3 \mod n$ meaning that $c_1 = -1 + 0 + 0 - 1 + 3.1 = 1$.
- $\mathcal{E}_{pk_B}(c_2) = \#NA$.

(l) A sets $\mathcal{E}_{pk_B}(c_i) := \mathcal{E}_{pk_B}(c_i)^{r_i} \mod n$ for all i where $r = 11 = (r_3r_2r_1r_0)_2 = (1001)_2$,

- $\mathcal{E}_{pk_B}(c_0) := \mathcal{E}_{pk_B}(c_0)^{r_0} \mod n = \mathcal{E}_{pk_B}(5)^1$ meaning that $c_0 := 5$.
- $\mathcal{E}_{pk_B}(c_1) := \mathcal{E}_{pk_B}(c_1)^{r_1} \mod n = \mathcal{E}_{pk_B}(1)^0$ meaning that $c_1 := 0$.
- $\mathcal{E}_{pk_B}(c_2) := \#NA$.

(m) A sends $\mathcal{E}_{pk_B}(c_i)$ to B in random order.

(n) B checks whether one of $\mathcal{D}_{sk_B}(\mathcal{E}_{pk_B}(c_i)) = 0$. Since it is so, B sets $\delta_B = 1$.

8. B computes $\mathcal{E}_{pk_B}(z \div 2^\ell) = \mathcal{E}_{pk_B}(z_{\ell+1}) = \mathcal{E}_{pk_B}(0)$ meaning that $(\ell + 1)$st bit of z is 1.

9. B sends $\mathcal{E}_{pk_B}(z \div 2^\ell) = \mathcal{E}_{pk_B}(1)$ and $\mathcal{E}_{pk_B}(\delta_B) = \mathcal{E}_{pk_B}(1)$ to A.
10. A sets $\mathcal{E}_{pk_B}((\beta < \alpha)) := \mathcal{E}_{pk_B}(\delta_B) = \mathcal{E}_{pk_B}(1)$ meaning that $(\beta < \alpha) = 1$.
11. A computes $\mathcal{E}_{pk_B}(\gamma) = \mathcal{E}_{pk_B}(z \div 2^\ell).\mathcal{E}_{pk_B}(r \div 2^\ell).(\mathcal{E}_{pk_B}((\beta < \alpha)))^{-1} \mod n^2 = \mathcal{E}_{pk_B}(1).(\mathcal{E}_{pk_B}(1)).(\mathcal{E}_{pk_B}(1))^{-1} \mod n^2 = \mathcal{E}_{pk_B}(1)$ meaning that $\gamma = 1$.
12. A sends $\mathcal{E}_{pk_B}(\gamma)$ to B.
13. B decrypts $\mathcal{E}_{pk_B}(\gamma)$ and gets $\gamma = 1$ and then encrypts it with pk_A and sends it to A back.
14. A decrypts $\mathcal{E}_{pk_A}(\gamma)$ and gets $\gamma = 1$.

Output: $\gamma = 1$ meaning that $3 \leq 5$.

Parameters:

$a = 3, b = 5$
$p = 7, q = 13, n = 91, n^2 = 8281$
$t = 2$
$r = 9 = (1001)_2, (r \div 2^\ell) = 1$
$z = 15 = (1111)_2, (z \div 2^\ell) = 1$
$\alpha = 1 = (001)_2$
$\beta = 7 = (111)_2$
$\tilde{\alpha} = 6 = (110)_2$
$w = (w_2 w_1 w_0) = (100)_2$
$\delta_A = 1, \delta_B = 1$
$s = -1$
$c_0 = 5, c_1 = 0, c_3 = none$
$\gamma = 1$

APPENDIX B

ASCII Character Codes

The first 32 characters in the ASCII-table are unprintable control codes and are used to control peripherals such as printers. Codes 32-127 are common for all the different variations of the ASCII table, they are called printable characters, represent letters, digits, punctuation marks, and a few miscellaneous symbols. We can find almost every character on the keyboard. Character 127 represents the command DEL [1].

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	32	20	Space	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	TAB	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SO	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	|
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Table B.1: ASCII Encoding Table

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name	**:**	Bektaş, Atilla
Nationality	**:**	Turkish (TR)
Date and Place of Birth	**:**	1978, Elazığ, Turkey
E-mail	**:**	bektasatilla@gmail.com

EDUCATION

Degree	**Institution**	**Year of Graduation**
M.S.	METU, IAM, Cryptography	2005
B.S.	METU, Mathematics	2001

PROFESSIONAL EXPERIENCE

Year	**Place**	**Position**
2013 - Present	Capital Markets Board	Manager
2002 - 2013	Capital Markets Board	Application Programmer
1999 - 2001	METU, Mathematics	Student Assistant