Doctoral Thesis

Cryptographic Protocols for Secure Electronic Commerce

Supervisor Professor Tatsuaki Okamoto

Department of Social Informatics
Graduate School of Informatics
Kyoto University

Takuho Mitsunaga

April, 2016

Cryptographic Protocols for Secure Electronic Commerce

Takuho Mitsunaga

Abstract

With the rapid growth in the Internet usage, opportunities for e-commerce transactions including online auctions, shopping and banking have been rapidly increasing. On the other hand, many cases of cyber attacks, as in intrusions into computers and networks resulting in a data corruption or defacement, information theft and system suspension, have been continuously observed. In this situation, realizing e-commerce transactions with enhanced safety and privacy is one of the important issues especially from cyber security perspectives.

This thesis focuses on online auctions as an example of safe e-commerce transactions since they require more complex procedures (protocols) compared to the other types of simple e-commerce transactions and have many issues to be solved in terms of safety and privacy. If online auctions can be safely realized, it can be a basis of technologies for various complex types of e-commerce transactions. One of the security challenges in online auctions is the risk of result manipulation by an auctioneer. In case of sealed bid auctions, where information on bidding prices is not disclosed to any bidders, it is possible for an auctioneer to manipulate a part of the bidding prices to derive an unfair result. In order to solve this problem, a cryptographic method called secure auction protocol is proposed. This is to conduct auctions with encrypted bidding prices, and this maintains bidders' privacy since the auctioneer can not obtain any information other than the results. As the method also allows bidders to verify the validity of the results, it can be said that it is a safe and reliable e-commerce transaction. This thesis proposes efficient secure auction protocols applying BGN encryption, which has homomorphic property, to "Bit Slice" method. Further expanding this proposal, an efficient protocol for M+1 price auction (an auction for M identical goods) is also proposed.

This thesis continues to provide observations on e-commerce mechanisms by implementing game theory which apply cryptography. Mechanism design based on game theory is used for constructing social rules to provide desirable results

for stakeholders by analyzing system design from broad perspectives. Protocols for optimized users' utilities based on the game theory are significant in efficiently realizing various complicated e-commerce transactions. Some previous researches suggest punishment strategies which provide a penalty to dishonest participants in order to prevent unfair transactions, however, it does not work in some cases. This thesis identifies such issues in the existing protocols and proposes a new method.

Finally, a secure method for website authentication is introduced. The majority of e-commerce transactions are realized through websites. Therefore, techniques to correctly authenticate users on web environment is crucial as a basic function for safe e-commerce transactions. Password authentication using a tuple of ID and password configured by each user is commonly applied for website user identification. Many users use the same password for different websites since it is difficult to configure and manage different sets of credentials, which results in damages caused by "list-based attacks": unauthorized login attempts with leaked credentials. Such unauthorized logins to websites have been continuously conducted causing monetary damages, and this is considered as one of the social issues. Therefore, a new authentication method which is more secure and useful than the password authentication is in great demand. This thesis proposes a web service authentication system based on public key authentication, which is resistant to the list-based attacks.

The series of research aims to realize secure e-commerce transactions using encryption.

安全な電子商取引のための暗号プロトコル

満永　拓邦

内容梗概

近年，インターネットの急速な普及に伴い，オンラインオークション，オンラインショッピングやオンラインバンキングなどの電子商取引が身近なものとなり，我々の生活に不可欠なものとなってきている．一方で，コンピュータやネットワークに不正に侵入し，データの破壊や改ざん，情報窃取，システム停止などの損害を与えるサイバー攻撃が後を絶たない．そのため安全性やプライバシを保持した電子商取引を実現することは大きな課題であり，サイバーセキュリティの観点でも最重要課題の一つと言える．

本論文では，まず初めに安全な電子商取引の一例として，オンラインオークションについて取り上げる．これは電子商取引の中でも，オークションは単純な商取引よりも複雑な手順（プロトコル）を必要とするため，安全性やプライバシ上の課題は多く，オークションを安全に実現することができれば，様々な複雑な形態の電子商取引を実現する基盤技術となるためである．オンラインオークションにおけるセキュリティ上の課題は，主催者によるオークション結果の改ざんである．参加者が相互に入札額を知ることができない封印入札方式では，主催者は参加者から収集した入札額を改ざんすることで主催者は不正な利益を得ることができる．この問題に対し暗号理論を用いて解決するセキュアオークションプロトコルという手法が提案されている．暗号化したままの入札額を用いてオークション結果を出力する手法であり，主催者がオークション結果以外の情報を得ることができないため，参加者のプライバシを保つことができる．また参加者が暗号化された入札額を用いて結果の正当性を検証することも可能であるため，参加者にとって信頼できる安全な電子商取引の一つであると言える．本論文では，ビットスライスと呼ばれる手法に対し，準同型性をもつ BGN 暗号を適用することで，効率的かつ安全な第一価格および第二価格オークションプロトコルを提案する．また提案方式を更に発展させ，M 個の財に対するオークション (M+1 価格オークション) に対しても，効率的な手法を提案する．

次に，電子商取引のメカニズムについて，暗号を応用したゲーム理論を用いて考察する．ゲーム理論にもとづくメカニズムデザインは，現実の制度設計を分析し，関係者にとって望ましい結果を与えるための社会的なルールの策定に

利用される．種々の複雑な電子商取引を効率的に実現するために，こうした考えにもとづく利得最適化のプロトコルは大変有益である．既存研究において，不正を行った参加者に対して罰を与えることで不正防止を図る Punishment Strategy が提案されている．既存方式では効果的に動かない事例を取り上げ，既存の方式の問題点を指摘するとともに，それを解消した新たな手法を提案する．

最後に，安全な Web サイトの認証方式を提案する．多くの電子商取引は，Web サイトを利用して実現されている．したがって，安全に電子商取引を行う基本機能として，Web 環境において利用者を正しく認証する技術は必須である．現在，Web サイトにおけるユーザ認証には，アカウント登録時にユーザが設定する ID とパスワードを用いるパスワード認証方式が一般的に利用されている．Web サイト毎に異なるパスワードを設定，管理することは容易ではないため，パスワードの使いまわしをするユーザも多数存在しており，漏えいした ID とパスワードの組み合わせを用いて不正にログインを試みるパスワードリスト型攻撃による被害発生に繋がっている．パスワードリスト型攻撃による不正ログインの被害は継続的に発生しており，金銭的な被害にも繋がるため社会的な問題となっている．そのため，パスワード認証より安全性が高く，利便性に優れる認証方式の普及は急務である．本論文では，パスワードリスト攻撃への耐性を持つ公開鍵認証方式に基づく Web サービス認証方式の提案を行う．

本論文では，サイバーセキュリティの様々な問題に対して暗号理論を用いて多面的な解決を試みており，これら一連の研究により，安全な電子商取引の実現を図る．

Cryptographic Protocols for Secure Electronic Commerce

Contents

Chapter 1 Introduction

1.1 Background

With the spread in the Internet usage, e-commerce transactions have also been developed. E-commerce transactions on the Internet has an advantage that it is open to any users without any limit in time or place, and its market has been increasing in recent years. Using e-commerce transactions on online shopping, auction, and banking on PC devices, tablets, and smartphones has become a part of everyday life. From economics perspectives, e-commerce transactions have been used in a larger scale year by year. According to the Organization for Economic Co-operation and Development (OECD), the definition of e-commerce transactions is classified into broad and narrow definition(see Table 1 for details) depending on the means [**?**]. A research conducted by the Ministry of Economy, Trade and Industry [**?**], following the broad definition, revealed that the e-commerce transaction market has grown from 34,560 billion yen in 2005 to 127,970 billion yen in 2015 as described in Figure 1. Since e-commerce transactions allow purchases beyond the borders, opportunities for international transactions have been increasing, and such market is expected to see a growth for the coming years. For e-commerce transactions, on online shopping sites for example, confidential data including credit card and bank account information are exchanged. Since these information require high confidentiality, they must be handled in a secure manner. On the other hand, however, the number of cyber attacks is increasing on the Internet, and their damage has been reported continuously. For example, their damage caused by unauthorized online bank transfers in Japan has been increasing, approximately from 300 million yen in 2011 to 3 billion yen in 2015 [**?**], and this is considered as a serious issue. According to Japan Consumer Credit Association [**?**], the cause of credit card abuses is mainly stealing information by cyber attacks etc. rather than forged credit cards by skimming. In fact, according to the aforementioned research by the Ministry of Economy, Trade and Industry, one of the highest priorities for users in choosing an e-commerce website is that the security measures are in place [**?**]. In this regard, it is necessary for e-commerce

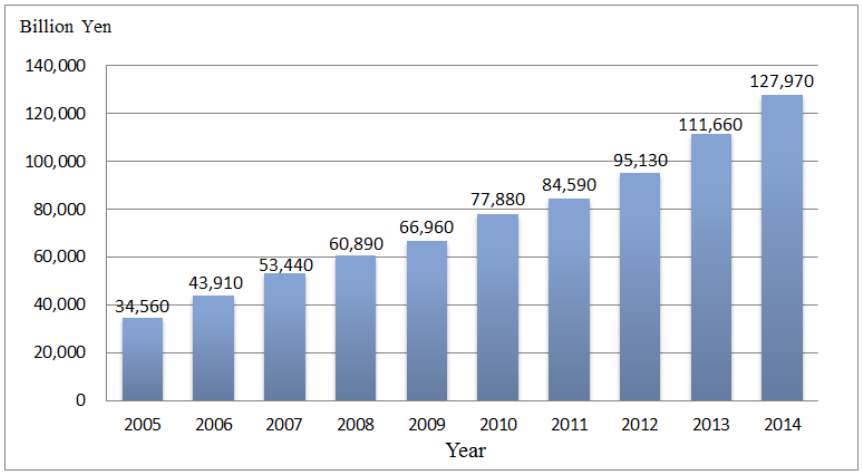

Figure 1: Increasing number of e-commerce transactions.

Table 1: The OECD definitions of e-commerce transactions and interpretation guidelines.

E-commerce transactions	OECD definitions
BROAD definition	An electronic transaction is the sale or purchase of goods or services, whether between businesses, households, individuals, governments, and other public or private organizations, conducted over computer-mediated networks. The goods and services are ordered over those networks, but the payment and the ultimate delivery of the good or service may be conducted on or offline.
NARROW definition	An Internet transaction is the sale or purchase of goods or services, whether between businesses, households, individuals, governments, and other public or private organizations, conducted over the Internet. The goods and services are ordered over the Internet, but the payment and the ultimate delivery of the good or service may be conducted on or offline.

service providers to prepare a platform (i.e. a website) which has appropriate security measures implemented. Considering the situation, it is safe to say that realizing secure e-commerce transactions has a significant value. This thesis aims to provide a model for secure e-commerce transactions by considering the topic from three different perspectives.

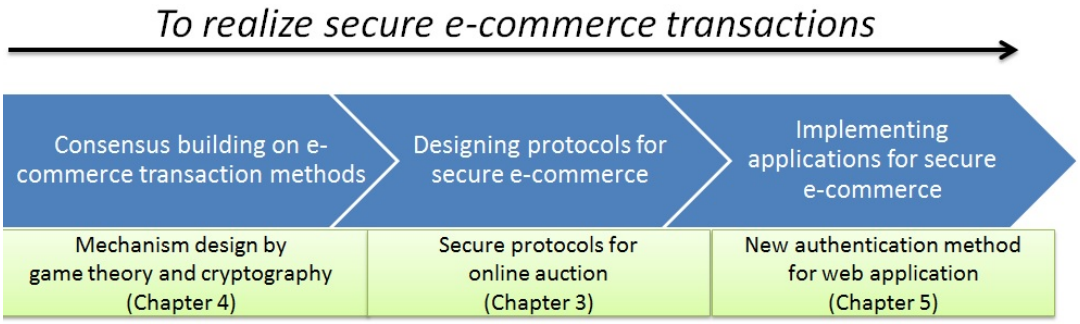

Figure 2: 3-layer approach for realizing secure e-commerce transactions.

1.2 This thesis's approach

In order to realize secure e-commerce transactions, this research takes a multi-layer approach, since one solution can not cover the whole security issues regarding to e-commerce transactions. Instead of focusing on a single circumstance, this research aims to achieve constructing 3-layers approach for secure e-commerce transactions and building a secure model for each layer. Being designed to be applicable for other areas as well, these models are expected to contribute to security for wider areas of e-commerce transactions by applying them.

1.2.1 Consensus building on e-commerce transaction methods

If there is a group of persons who wish to launch an e-commerce transaction, they need to reach an agreement on rules and formats for conducting such transactions. In terms of format, for example, there are various ways such as sales contracts, rental contracts, first-come, first-serve basis or auction, and some discussions take place in order to make a decision. However, such a consensus building process has difficulties and sometimes time-consuming. This is because, particularly e-commerce transactions involve diverse stakeholders' profits, and therefore they are likely to behave based on different interests. When multiple stakeholders are involved, the difficulties for reaching an agreement increase, as described in Byzantine Generals Problem [?, ?, ?]. Some researches have been done to solve such challenges by applying cryptographic theory for smooth consensus building process [?, ?] . This approach is thought to be effective since

cryptographic theory can provide legitimate users with reliable information that is necessary for decision making. Nevertheless, there still may be dishonest players, and the possibility can not be completely eliminated. In case where any dishonest action is detected, "Punishment Strategy" was proposed to disincentive such a dishonest action. This can potentially contribute in constructing a better system which can satisfy each stakeholder. This thesis discusses improvement on a punishment strategy which applies cryptographic theory in order to create a protocol to further maximize players' profit.

1.2.2 Designing protocols for secure e-commerce transactions

As a next step, based on the rules decided through the consensus building, the possibility to construct such an e-commerce transaction system needs to be considered. In order for the participants to use e-commerce transactions without concern, dishonest actions by administrators or other participants need to be detected, and user privacy has to be protected. Therefore, this layer should consider system design for privacy of the participants' behavior and protocols to detect participants' dishonest activities. In this layer, we use encryption protocols to to achieve the challenges. Furthermore, ideally, a model e-commerce transaction involves complicated protocols rather than the simple ones. This thesis focuses on auctions as a model for e-commerce transactions. This is because it involves complex issues such as user privacy and administrator's dishonest actions, and thus the protocols used for auctions can be applied in the other areas as well.

1.2.3 Implementing application secure e-commerce transactions

Last but not least, in order to realize secure e-commerce transactions, system security needs to be considered. Even with an e-commerce transaction algorithm which is securely designed in theory, it is not entirely secure if there are some security issues in its usage. In reality, even with secure protocols, the problems such as information theft and an unauthorized operation can occur due to some technical issues or faults by the administrators or participants. This is thought to be a critical issue in e-commerce transactions and needs to be resolved. This thesis, based on the fact that e-commerce transactions are mostly conducted on websites, focuses on the system security on websites. Threats on website sys-

Threats	Countermeasures	
	Developers	Users
Defacement/Data leakage	- Secure coding - Secure configuration	-
(D)DoS	- Server resource management - Network redundancy	-
Unauthorized login	- ID/Password management	

Table 2: Main web-base threats and examples of countermeasures.

tems include (Distributed) Denial of Service, unauthorized login, defacement, and data leakage. Table 2 describes where a developer and a user stand for each of the threat. For (D)Dos mitigation, resource distribution and network redundancy have been already researched [?, ?, ?], and those devices for (D)Dos mitigation to reduce the risks are being provided. For the secure coding and configuration to prevent defacement or data leakage, security related organizations such as OWASP and IPA have been publishing information available for developers [?, ?, ?, ?]. On the other hand, ID/Password management as a mitigation against unauthorized login is not an issue for the developers but for users. This is a complex issue that cannot be solved easily in the perspective of the balance between usability and security. Fast IDentity Online Alliance (FIDO) has been conducting research on new authentication technologies aiming for passwordless world [?, ?]. However, technologies used in this movement need new devices for authentication such as a fingerprint reader. Considering these issues, this thesis focuses on ID/Password management and add some other techniques to it in order to propose a securer authentication protocol.

1.3 Related Works

Recently, as the Internet has expanded, many researchers have become interested in secure auction protocols and various schemes have been proposed to ensure the safe transaction of auction since it requires more complex procedures (protocols) compared to other types of simple commerce transactions and has

many issues to be solved in terms of safety and privacy. If online auction can be realized safely, it can be a basis of technology for various complex types of e-commerce transactions. A secure auction is a protocol in which each player can find only the highest bid and its bidder (called the first price auction) or the second highest bid and the first price bidder (called the second price auction) [?, ?, ?, ?]. A simple solution is to assume a trusted auctioneer. Bidders encrypt their bids and send them to the auctioneer, and the auctioneer decrypts them to decide the winner. To remove the trusted auctioneer, some secure multi-party protocols have been proposed. The common essential idea is the use of threshold cryptosystems, where a private decryption key is shared by the players. Jakobsson and Juels proposed a secure MPC protocol to evaluate a function comprising a logical circuit, called mix-and-match [?]. As for a target function f and a circuit that calculates f, C_f, all players evaluate each gate in C_f based on their encrypted inputs and the evaluations of all the gates in turn lead to the evaluation of f. Based on the mix-and-match protocol, we can easily find a secure auction protocol by repeating the millionaires' problem for two players [?, ?]. However, the mix-and-match protocol requires two plaintext equality tests [?, ?] for a two-input one-output gate. And one plaintext equality test requires one distributed decryption among players. Thus, it is important to reduce the number of gates in C_f to achieve function f. Kurosawa and Ogata suggested the "bit-slice auction", which is an auction protocol that is more efficient than the one based on the millionaire's problem [?]. Boneh, Goh and Nissim suggested a public evaluation system for 2-DNF formula based on an encryption of Boolean variables [?]. Their protocol is based on Paillier's scheme [?], so it has additive homomorphism and in addition the bilinear map allows one multiplication on encrypted values. As a result, its property allows the evaluation of multivariate polynomials of a total of degree two on encrypted values. In this thesis, we introduce bit-slice auction protocols based on the public evaluation of the 2-DNF formula. For the first price auction, the protocol uses no mix-and-match gates. For the second price auction, we use the mix-and-match protocol fewer times than that suggested in [?]. There are many auction protocols [?, ?, ?], however, they have problems such as those described

hereafter. The first secure auction scheme proposed by Franklin and Reiter [**?**] does not provide full privacy, since at the end of an auction players can know the other players' bids. Naor, Pinkas and Sumner achieved a secure second price auction by combining Yao's secure computation with oblivious transfer assuming two types of auctioneers [**?**]. However, the cost of the bidder communication is high because it proceeds bit by bit using the oblivious transfer protocol. Juels and Szydlo improved the efficiency and security of this scheme with two types of auctioneers through verifiable proxy oblivious transfer [**?**], which still has a security problem in which if both auctioneers collaborate they can retrieve all bids. Lipmaa, Asokan and Niemi proposed an efficient $M+1st$ secure auction scheme [**?**]. In this scheme, the trusted auction authority can know the bid statistics. Abe and Suzuki suggested a secure auction scheme for the $M+1st$ auction based on homomorphic encryption [**?**]. The $M+1st$ price auction is a type of sealed-bid auction for selling M units of a single kind of goods, and the $M+1st$ highest price is the winning price. M bidders who bid higher prices than the winning price are winning bidders, and each winning bidder buys one unit of the goods at the $M+1st$ winning price. However, in their scheme, a player's bid is not a binary expression. So, its time complexity is $O(m2^k)$ for a m-player and k-bit bidding price auction. Tamura, Shiotsuki and Miyaji proposed an efficient proxy-auction [**?**]. This scheme only considers the comparison between two sealed bids, the current highest bid and a new bid. However, this scheme does not consider multiple players because of the property of the proxy-auction. Some researches are conducted based on secret sharing for secure auction protocols [**?**, **?**].

In chapter 4, we provide observations on e-commerce mechanisms by implementing game theory which applies cryptography. Mechanism design based on game theory is used for constructing social rules to provide desirable results for stakeholders by analyzing actual system design from broad perspectives. Protocols for optimized users' utilities based on the game theory are significant in efficiently realizing various complicated e-commerce transactions.

Some previous research suggests punishment strategies which provide penalty to dishonest participants in order to prevent unfair transactions, however, it

does not work efficiently in some occasions. One of the most important ideas in game theory is equilibrium in which it is the best way for all player to follow actions. Two kinds of equilibrium were proposed. First, Nash equilibrium (named after John Forbes Nash, who proposed it) is a solution concept of a game involving two or more players, in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by only changing its own strategy [**?**, **?**]. The other is a correlated equilibrium which was proposed by Robert Aumann [**?**] and is a solution concept that is more general than the well known Nash equilibrium. The idea is that each player chooses its action according to its observation of the value of the single public signal. That signal is supposed to be sent by a trusted third party called a mediator. It chooses the set of moves according to the right joint distribution and privately tells each player what its designated move is. Then the next question is "can we remove the mediator by using some protocols". In the case of a two-player game, it is well known that in the standard cryptographic models the answer is positive, provided that the two players can interact [**?**]. This positive result can be carried over to the game theory model as well. Specially, we consider an extended game, in which the players first exchange some messages (this part is called "cheap talk" in game theory), and then choose their actions and execute them simultaneously as in the original game. In [**?**], they suggested the concept of punishment strategy which is a kind of rule for players not to abort in the cheap talk phase. If a player aborts, the other players take actions that lead aborting player's utility low. So all player do not have incentive to abort in the cheap talk phase and deviating from the action in the original game. Matsubara proposed a new mechanism for auction contracts to solve problems such as free-riding with digital goods on peer-to-peer network service [**?**]. With his mechanism, it is guaranteed that each user reveals its true information in a single good case. To reduce risky situation such as a fraud in exchange processes of e-commerce, Matsubara and Yokoo developed fraud-free exchange mechanisms [**?**]. Their research focused on an entry fee since the risk of frauds is affected by a cost to join the e-commerce networks. First mechanism reduces the entry fee by integrating multiple deals and controlling goods and

money flows. The other reduces the entry fee by incorporating a third party into the exchange process. A punishment strategy is proposed to lead players to appropriately behave by posing a penalty to dishonest players. We show an example of game in which a punishment strategy does not work and suggest an improved definition of a punishment strategy.

In chapter 5, we will propose a new user identification system for web services based on public keys which are resistant to list-based attacks. On web services which require user identification, user account registration is needed upon using the services. Most commonly, password authentication with a pair of credentials is used for this purpose. In many web services, it is required to create complex passwords with more than a certain number of letters, which is hard to guess for attackers. To detect unauthorized login attempts for web service, researches regarding anomaly detection to are conducted [**?**, **?**]. Basic idea of anomaly detection techniques is to divide user's activities into a normal behavior and an abnormal behavior(e.g. a login failure). If the amount of abnormal behaviors exceeds a threshold level that is set beforehand, it is recognized as attacks against web service. Anomaly detection is effective when abnormal behaviors can be identified. However, in recent years, more sophisticated attacks such as list-based attacks are observed [**?**]. Pairs of credentials which seem to be leaked from a web service were made available in black markets, and list-based attacks, where attackers abuse these credentials to log in to different web services, have been conducted. In this case, even if the passwords are complex enough, unauthorized login may occur in different web services if the users share the same password in some services, as attackers can use correct credentials that are already leaked. As countermeasures against list-based attacks, an alternative authentication method such as to a two-factor authentication or a client certification is effective. The Two-factor authentication system is applicable for some web services. With the increasing share of mobile devices, there are researches using mobile devices as the factor of authentication [**?**, **?**, **?**]. As for a client certification, Kobayashi and others proposed an authentication system based on client certification and FakeBasic [**?**]. However, since its configuration and usage flow are complicated, it does not seem to be commonly employed.

Chapter 2 Preliminaries

This chapter introduces basic ideas on cryptography and game theory used in this thesis.

2.1 Public key encryption

With the spread of the Internet, there are increasing needs for secure communication protocols using encryption among remote users. Public key encryption is widely applied in Internet-related techniques such as SSL (Secure Socket Layer), since it does not require key delivery before use, differently from common key encryption (symmetric key encryption). Public key encryption uses two types of keys: a public key (a key for encryption) and a secret key (a key for decryption). A user who wishes to have encrypted communication generates a pair of keys and puts the public key in a place which is accessible to the other users, and keeps the secret key securely managed. The others can generate an encrypted message using the public key and send to the user who has put the public key. The user who receives the message can decrypt the message by using the secret key. Compared to common key encryption, which requires a key delivery among users, this method can be deployed easier in key management since public keys can be shared easily on the Internet. Public key encryption uses the following three algorithms: key generation algorithm, encryption algorithm and decryption algorithm. A key generation algorithm is the one for preparation, and the users who wishes to receive encrypted messages needs to execute the algorithm beforehand. The key generation algorithm outputs the user's public and secret keys. A public key is used to generate encrypted messages, and a secret key is used to retrieve the original message. When executing the key generation algorithm, the user inputs a value called security parameter into the algorithm. Security parameter indicates how difficult it would be to break the security of the encryption system itself such as decrypting encrypted messages without the secret key. A random number is also input into the key generation algorithm. Since the algorithm chooses a different random number for each time it is executed, an individual pair of public and secret key is as-

signed for each user. In order to send an encrypted message, senders input the message and the receiver's public key, then execute the encryption algorithm (since public keys are public information, senders have access to receiver's the public key). A receiver inputs their own secret key and the encrypted message into the decryption algorithm and retrieves the original message. Here below explains the algorithms. Let m, sk, pk, $KeyGen()$, $Enc()$, $Dec()$ be a message, a secret key, a public key, a key generation algorithm, an encryption algorithm and a decryption algorithm respectively.

1. A receiver use $KeyGen()$ to generate sk and pk with appropriate security parameter. The receiver put pk in a place where potential senders can access.
2. A sender obtains a receiver's the public key pk.
3. The sender encrypts a message m using $Enc()$ and pk and generates an encrypted message $c = Enc(pk, m)$.
4. The sender sends c to the receiver.
5. The receiver decrypts c using $Dec()$ and the secret key sk, and retrieves the message m. $(m = Dec(sk, c))$

Even if the communication channel is eavesdropped, m cannot be obtained from c, and the confidentiality of c is guaranteed.

2.2 Digital signature

Digital signature is a technique to prevent impersonation and data manipulation by using public key cryptography. In the real world, to prove a document was generated or approved by a specific user, he or she puts a signature or a stamp on the document. However, such signatures cannot be provided on digital documents and do not have credibility since digital data can be easily copied and pasted. It also has a risk that data is manipulated through insecure communication channels. Digital signature based on public key encryption is an effective method to reduce such risks. Digital signature uses the following three algorithms: a key generation algorithm, a signing algorithm and a verification algorithm. Each user performs the key generation algorithm and stores their secret key in a secure place, while the public key is set open to public. A user (a

signer) uses the signing algorithm and a message to create a digital signature on the message with the secret key. For digital signature, a public key and a secret key are also referred to as a verification key and a signing key. A user (a verifier) who received the message and the digital signature can verify if the signature is authentic by inputting them into the verification algorithm. A verifier also inputs the signer's (who is supposed to have signed the document) public key. Since public keys are public information, verifiers have access to the public key of the signer. Here below explains the algorithms. Let m, sk, pk, $KeyGen()$, $Sign()$, $Verify()$ be a message, a secret key, a public key, a key generation algorithm, a signing algorithm and a verification algorithm respectively.

1. A signer use $KeyGen()$ to generate sk and pk with appropriate security parameter. The signer put pk in a place where potential verifiers can access.
2. A verifier obtains the signer's public key pk.
3. The signer obtains a signature s from a message m by using his/her own secret key sk. $(s = Sign(sk, m))$
4. The signer sends m and s to a verifier.
5. The verifier verifies the signature s by using the public key pk and the original message m.$(Verify(pk, s, m) = 0/1)$

Even if the communication channel is eavesdropped, s is obtained by malicious users, they can not generate digital signature s' for the other message m', then there is no risk of impersonation.

2.3 Public Key Infrastructure

As an infrastructure for secure e-commerce transactions based on such public key encryption, Public Key Infrastructure (PKI) is used. To verify a public key holder (who has the secret key corresponding to the public key), an "electronic certificate" serves, and organizations which issue such certificates are called " certification authorities ". For example, if a user receives a public key (a certificate that includes public key data) from another user in order to communicate using PKI, he/she needs to check if there is no flaw in the certificate or the issuing authority is reliable. As a next step, the user needs to verify if the certificate was indeed issued by the certification authority which is indicated

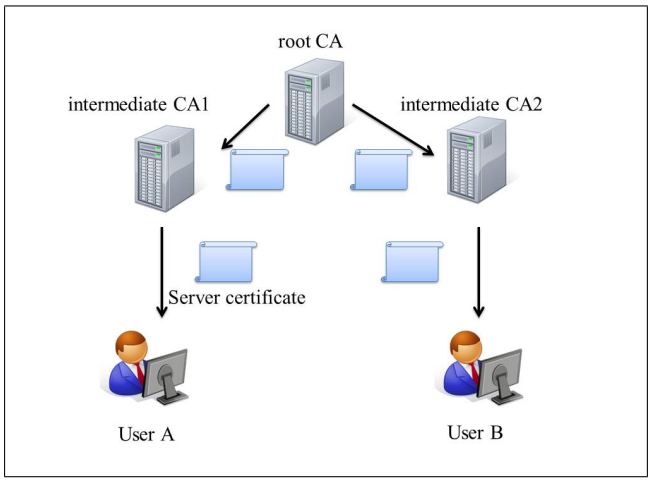

Figure 3: Layer structure of CA.

in the certificate. For that purpose, the signature of the indicated certification authority needs to be verified. In order to verify the signature, the public key of the certification authority is used. There are two ways for a certification authority to issue a certificate.

1. The certification authority itself, or
2. Its superior certification authority, if any

Even for the second method, a certification authority which issued the certificate as in the first method can be found. Such a certification authority is called "root certification authority". Certificates issued by a root certification authority has a signature of the certification authority itself. These certificates are called "self-signed certificates". Let us assume a case where user A has a certificate issued by certification authority CA1 and user B has one by CA2. If those two certificate authorities are certified by the same root certification authority (hence, receive certificates from the same root certification authority), both user A and B are to trust the root certification authority. As a result, user A and B can establish a trust relationship through the root certification authority. For those certification authorities who have a mutual trust relationship, its individual certificate holders also have a trust relationship. When the user A receives a certificate (the public key) of user B, A also needs the certificate of CA2 for verification. In order to verify that the certificate issued by CA2 is provided by a root certificate authority which user A trusts, A first needs to verify

the signature on CA2's certificate. Also to verify that user B's certificate was indeed issued by CA2, the signature on B's certificate needs to be verified. As a basis of trust, certification authorities play an important role. For example, a certification authority's key serves to provide a signature on certificates. However, if the key is stolen, certificates can be issued as much as the stealers want by impersonating the authority. For this reason, certification authorities need to securely protect their secret keys. There are various types of PKI as well: GPKI for government bodies and UPKI for universities and academic bodies etc. GPKI consists of bridge certification authorities and those belonging to ministries and agencies. It was constructed with an aim to process paperless administrative work on the Internet, which includes applications and reports submitted by citizens and notification provided by ministries and agencies [**?**]. The system is used for the authority to verify that such an application or notification was indeed created by the person or the organization indicated in the document, and that the contents were not modified by a third party. UPKI was established in order for research institutes to collaboratively utilize academic information resource (super computers, electronic contents, network etc.) in a secure and useful way [**?**, **?**]. This system is managed by National Institute of Informatics (NII).

2.4 Homomorphic Encryption

Since cloud service has been used more widely, its cost has been lowered. The service has enabled wider function as to process complicated and mass operation in order to obtain useful information from the collected data. It is expected that the cloud will evolve into a valuable tool in the area of information processing. On the other hand, there has been much concern about applying the cloud in the business environment. Especially for corporate use, confidentiality of the data on the cloud is an issue to be considered. Information management at corporate environment should be thoroughly and carefully discussed since information leakage, especially customer's personal information, can lead to a fatal issue to a company. Therefore, for corporate use, security measures are necessary when storing data in the cloud, and data encryption is one of the

useful measures. The confidentiality of the data is guaranteed by encrypting the data and properly managing the keys, and such measures are thought to be valid in terms of data protection. However, when utilizing the data stored in the cloud to the full extent, it is difficult to process the encrypted data since statistical analysis and searching require the original data. One of the simple methods for analysis processing on the cloud is to decrypt the encrypted data within the cloud and process the original data. However, security measures for the systems (key management, cloud access restriction etc.) is required for this operation, which would be a disadvantage on the cost. Therefore, this method does not seem to be a useful solution. To break through for the above issue, the following techniques to protect privacy are being discussed: homomorphic encryption, which enables processing encrypted data, and Multi Party Calculation (MPC), where multiple calculators collaboratively process data following secure procedures while keeping the data concealed. As an example, using homomorphic encryption, it is possible to compile encrypted examination results while the data is concealed and derive the compile results only. This includes RSA encryption [**?**] which can multiply encrypted numbers, and additive ElGamal encryption [**?**] and Paillier encryption [**?**] which can add up encrypted numbers. These encryption methods can be applied to electronic voting and electronic cash, however, the usage is limited since it only allows encrypted multiplication or addition. In order to address these issues, Gentry proposed the detailed construction method for fully homomorphic encryption in 2009 [**?**], which theoretically allowed processing arbitrary logical operation while encrypted. Nonetheless, fully homomorphic encryption has an issue with its processing cost and encrypted data size, which needs to be solved upon implementation. On the other hand, applied research for SHE homomorphic encryption, which is a basic component of fully homomorphic encryption, has been drawing attention these years. Although there is a limit in number of times to perform operations, this encryption allows both encrypted addition and multiplication. SHE homomorphic encryption has a limit in the number of times to perform for encrypted operations, but both processing cost and encrypted data size are quite small. This would be an advantage in practical use

in diverse areas.

- **Additively Homomorphic Encryption**
  This encryption only allows encrypted addition and includes Paillier encryption and additive ElGamal encryption. It has the same level of processing capacity as in RSA encryption and is expected to be applied in electronic voting and electronic cash.

- **Somewhat Homomorphic Encryption (SHE)**
  This encryption allows both encrypted addition and multiplication within a limited number of times of operation. Although it has a limited features compared to Fully Homomorphic Encryption, its low processing cost and small encryption data size can be an advantage when applied in various statistical processing.

- **Fully Homomorphic Encryption (FHE)**
  This encryption allows arbitrary operation. Since Gentry proposed the construction method of FHE using ideal lattice in 2009, application and construction method of FHE has been rapidly conducted. For realizing practical processing capacity and encryption data size, further improvement and research is required.

In this section we first recall the somewhat homomorphic encryption scheme published by van Dijk, Gentry, Halevi and Vaikuntanathan [**?**]. The scheme is based on a set of public integers: $x_i = p \cdot q_i + r_i$, $0 \leq i \leq \tau$, where the integer p is secret. We use the same notation as in [x]. For a real number x, we denote by $\lceil x \rceil$, $\lceil x \rfloor$ and $\lfloor x \rfloor$ the rounding of x up, down, or to the nearest integer. We denote $[z]_p$ by z mod p. For a real number z and an integer p we denote the reduction of z modulo p by $[z]_p$ with $-p/2 < [z]_p \leq p/2$. We write $f(\lambda) = \mathcal{O}(g(\lambda))$ if $f(\lambda) = \mathcal{O}(g(\lambda) log^k g(\lambda))$ for some $k \in \mathbb{N}$.

The scheme parameters. Given the security parameter λ, the following parameters are used:

- γ is the bit-length of the x_i's.
- η is the bit-length of secret key p.
- ρ is the bit-length of the noise r_i.
- τ is the number of x_i's.
- ρ' is a secondary noise parameter used for encryption.

For a specific η-bit odd integer p, we use the following distribution over γ-bit integers:

$\mathcal{D}_{\gamma,\rho}(p) = \{Choose\ q \leftarrow \mathbb{Z} \cap [0, 2^\gamma/p), r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$: $Output\ x = q \cdot p + r\}$

$KeyGen(1^\lambda)$. Generate a random odd integer p of size η bits. For $0 \leq i \leq \tau$ sample $x_i \leftarrow \mathcal{D}_{\gamma,\rho}(p)$. Relabel so that x_0 is the largest. Restart unless x_0 is odd and $[x_0]_p$ is even. Let $pk = (x_0, x_1, \ldots, x_\tau)$ and $sk = p$.

$Encrypt(pk, m \in \{0, 1\})$. Choose a random subset $S \subseteq \{1, 2, \ldots, \tau\}$ and a random integer r in $(-2^{\rho'}, 2^{\rho'})$, and output the ciphertext:

$$c = \left[m + 2r + 2 \sum_{i \in S} x_i \right]_{x_0}$$

$Evaluate(pk, C, c_1, \ldots, c_t)$. Given the circuit C with t input bits, and t ciphertexts c_i, apply the addition and multiplication gates of C to the ciphertexts, performing all the additions and multiplications over the integers, and return the resulting integer.

$Decrypt(sk, c)$. Output m $\leftarrow$ (c mod p) mod 2. Note that since c mod $p = c - p \cdot \lfloor c/p \rceil$ and p is odd, one can compute instead: $m \leftarrow [c]_2 \oplus [\lfloor c/p \rceil]_2$. Applying Gentry's bootstrapping technique to the scheme above, a fully homomorphic property can be obtained. FHE and SHE are a useful tool for secure e-commerce transactions since they can provide multiplication or addition on encrypted data in the cloud. However, they have a issue to be solved regarding to computation costs. Even though efficient fully homomorphic schemes are introduced in recent research [**?**, **?**], they still require more computation costs and resources than previous homomorphic scheme such as BGN encryption [**?**]. Therefore in this thesis, we build up secure e-commerce transaction with BGN encryption scheme instead of FHE or SHE scheme.

2.5 Key sharing

In [**?**], efficient protocols are presented for a number of players to jointly generate an RSA modulus $N = pq$ where p and q are prime, and each player retain a share of N. In this protocol, none of the players can know the factorization of N. They then show how the players can proceed to compute a public exponent e and shares of the corresponding private exponent. At the end of the computation the players are convinced that N is a product of two large primes by using zero-knowledge proof. Their protocol was based on the threshold decryption that m out of m players can decrypt the secret. The cost of key generation for the shared RSA private key is approximately 11 times greater than simple RSA key generation. However, the cost for computation is still practical. We use this protocol to share private keys among auction managers.

2.5.1 Overview

We give an overview of the key generation protocol. The k parties wish to generate a shared RSA key. That is, they wish to generate an RSA modulus $N = pq$ and a public/private pair of exponents e, d where $e \cdot d = 1 \bmod \varphi(N)$. The factors p and q should be at least n bits each. At the end of the computation N and e are public, and d is shared between the k players in a way that enables threshold decryption and signatures. All players should be convinced that N is a product of two primes, but no coalition of at most $t = \lfloor \frac{k-1}{2} \rfloor$ players should have any information about the factors of N.

1. pick candidates: The following two steps are repeated twice.
   (a) secret choice: Each player i picks a secret n-bit integer p_i and keeps it secret.
   (b) trial division: Using a private distributed computation the k players determine that $p = p_1 + ... + p_k$ is not divisible by any prime less than some bound B_1. If this step fails repeat Step(a).

   Denote the secret values picked at the first iteration by $p_1, ..., p_k$, and at the second iteration by $q_1, ..., q_k$.
2. compute N: Using a private distributed computation the k players compute

$$N = (p_1 + ... + p_k) \cdot (q_1 + ... + q_k)$$

Other than the value of N, this step reveals no further information about the secret values $p_1, ..., p_k$ and $q_1, ..., q_k$.

3. biprimality test: The k players engage in a private distributed computation to test that N is the product of two primes. If the test fails, then the protocol is restarted from Step 1. We note that the biprimality test protocol is $k-1$ private and applies whenever two (or more) players are involved.
4. key generation: Given a public encryption exponent e, the players engage in a private distributed computation to generate a shared secret decryption exponent d.

2.5.2 Distributed biprimality test

We begin the detailed discussion of the protocol with the distributed biprimality test proposed in [**?**]. Player i has two secret n-bit integers p_i, q_i. All players know N where $N = pq = (\sum p_i)(\sum q_i)$. They wish to determine if N is the product of two primes without revealing any information about he factors of N. We refer to this test as a distributed biprimality test. The biprimality test is a probabilistic test carried out in both $\mathbb{Z}_N^*$ and a quadratic extension of $\mathbb{Z}_N^*$.

Throughout the section we are assuming that $p \equiv q \equiv 3 \bmod 4$(hence the resulting $N = pq$ is a Blum integer). This can be arranged ahead of time by having party 1 picks shares $p_1 \equiv q_1 \equiv 3 \bmod 4$. All the other players pick shares $p_i \equiv q_i \equiv 0 \bmod 4$.

Before describing the test we briefly discuss the structure of the quadratic extension of $\mathbb{Z}_N^*$ we will be using. We will be working in the group $\mathbb{T}_N = (\mathbb{Z}_N[x]/(x^2+1))^*/\mathbb{Z}_N^*$. Suppose all prime factors of N are equal to 3 mod 4. In this case, x^2+1 is irreducible in $\mathbb{Z}_N$ and $\mathbb{Z}_N[x]/(x^2+1)$ is a quadratic extension of $\mathbb{Z}_N$. A linear polynomial $f(x) = \alpha x + \beta$ in $\mathbb{Z}_N[x]/(x^2+1)$ is invertible if and only if gcd $(\alpha, \beta, N) = 1$. It follows that elements of $\mathbb{T}_N$ can be viewed as linear polynomials $f(x) = \alpha x + \beta$ in $\mathbb{Z}_N[x]$ with $\gcd(\alpha, \beta, N) = 1$. Two linear polynomials $f, g \in \mathbb{Z}_N[x]$ represent the same element of $\mathbb{T}_N$ if $f = \alpha g$ for some $\alpha \in \mathbb{Z}_N^*$.

Distributed biprimality test:

Step 1: The players agree on a random $g \in Z$. The value g is known to all the k players.

Step 2: Player 1 computes the Jacobi symbol of g over N. If $(\frac{g}{N}) \neq 1$ the protocol is restarted at Step(1) and a new random g is chosen.

Step 3: Otherwise, player 1 computes $v_1 = g^{(N-p_1-q_1+1)/4} \bmod N$. All the other players compute $v_i = g^{-(p_i+q_i)/4} \bmod N$. The players use the Benaloh's protocol [**?**] to compute $v = \Pi_{i=1}^{k} v_i \bmod N$. They then check if

$$v = \Pi_{i=1}^{k} v_i = \pm 1 \pmod{N}$$

If the test fails N is rejected. Otherwise they declare success.

2.5.3 Shared generation of public and private keys

Once the players successfully construct an RSA modulus $N = pq = (\sum p_i)(\sum q_i)$ they may wish to compute shares of $d = e^{-1} \bmod \varphi(N)$ for a given encryption exponent e.

Throughout this subsection, we set $\varphi = \varphi(N)$. Since e is an RSA exponent we know that $\gcd(e, \varphi(N)) = 1$. Recall that the public modulus $N = (\sum p_i)(\sum q_i)$ satisfies $\varphi(N) = \varphi = \sum_{i=1}^{k} \varphi_i$ where $\varphi_1 = N - p_1 - q_1 + 1$ and $\varphi_i = -p_i - q_i$ for $i = 2, ..., k$. Observe that for all $i = 1, ..., k$ player i can locally compute φ_i.

To compute shares of d the players must invert e modulo $\sum \varphi_i$ without exposing their φ_i. Unfortunately, traditional inversion algorithms, e.g. extended gcd, involve computations modulo $\sum \varphi_i$. We do not know how to efficiently perform modular arithmetic when the modulus is shared among the players. Fortunately, there is a trick for computing $e^{-1} \bmod \varphi$ without using any reductions modulo φ. We compute the inverse of $e \bmod \varphi$ in three steps:

1. Compute $\zeta = \varphi^{-1} \bmod e$.
2. Set $T = -\zeta \cdot \varphi + 1$. Observe that $T \equiv 0 \bmod e$.
3. Set $d = T/e$. One can easily verify that $d \equiv e^{-1} \bmod \varphi$ since $d \cdot e = T \equiv 1 \bmod \varphi$.

Using this observation there is no need for reductions modulo φ. Both methods rely on this observation.

The method to generate shares of d for public exponent e works as below. It is very efficient, but leaks $2\log k$ bits. This information can not help an opponent since it can be easily guessed.

Rather than exposing $\varphi \bmod e$ and then inverting it, we introduce how to

invert φ mod e while it is shared among the players. As a result, no information about φ is revealed. The protocol is $\lfloor \frac{k-1}{2} \rfloor$ private.

Step 1: Each player i picks a random $r_i \in \mathbb{Z}_e$.

Step 2: Players compute $\psi = (\sum r_i) \cdot (\sum \varphi_i) \bmod e$. At the end of the computation ψ is known to all players. If ψ is not invertible modulo e, the protocol is restarted at Step 1.

Step 3: Each player locally computes $\zeta_i = r_i \psi^{-1} \bmod e$. Observe $\sum \zeta_i = (\sum r_i)\psi^{-1} = \psi^{-1} \bmod e$. Hence, the players share $\varphi^{-1} \bmod e$ withiout revealing any information about their secret shares.

Step 4: Next, the players agree on a prime $P > 2Ne$. They wiew the shares $0 \leq \zeta_i \leq e$ as elements of $\mathbb{Z}_p$ and compute an additive sharing,

$$\textstyle\sum_{i=1}^{k} T_i = -(\sum \zeta_i)(\sum \varphi_i) + 1 \pmod{P}$$

Each player has a $T_i \in \mathbb{Z}_P$ and any minority of players learns no other information.

Step 5: From here, we regard the each value T_i as an integer $0 \leq T_i < P$. Our objective is to ensure that over the integers,

$$\sum_{i=1}^{k} T_i = -(\sum_{i=1}^{k} \zeta_i)(\sum_{i=1}^{k} \varphi_i) + 1 \tag{1}$$

We know that at the end of Step 4 we have $\sum T_i < kP$. Therefore, $\sum T_i = sP + u$ where $s < k$ and $0 \leq u < P$. Given a candidate value of $s \in [0,k)$ player 1 can set $d_1 \leftarrow d_1 - sP$. If the given s is correct then $0 \leq \sum T_i < P$ and the above equality holds over the integers. To determine the correct s the protocol proceeds to Step 6 with each possible value of s until the trial decryption in Step 6 succeeds.

Step 6: Assuming equality (1) holds over the integers, we know that e divides $\sum T_i$.

$$\textstyle\sum T_i = -(\sum \zeta_i)(\sum \varphi_i) + 1 = -(\sum \varphi_i)^{-1}(\sum \varphi_i) + 1 \equiv 0 \pmod{e}$$

Therefore, $d = (\sum T_i)/e$. Each player i now sets $d_i = \lfloor T_i/e \rfloor$. As a result we have $d = \sum d_i + r \bmod \varphi(N)$ where $0 \leq r \leq k$. player 1 can determine the value of r by trying all possible values fo $0 \leq r \leq k$ during a trial decryption.

2.6 Game theory

This subsection provides explanation on the basics of game theory. Game theory is a theory to consider decision making process in various situations involving multiple players, which can be used as analytical tools designed to help us understand the phenomena that we observe how players interact. This includes choosing an option out of multiple choices, and in particular, it is useful in decision making process where choices made by other players (including coincidence) have influence on the outcome. This theory helps in decision making process for the player itself, but also in predicting the possible outcomes in a situation where there are multiple players acting at their own will. Furthermore, the thesis can be applied in constructing better social rules; an auction protocol is one of the examples. Some basic terms used in game theory are as follows.

- **Player**: A body to make decision. Multiple players are involved in a situation.
- **Action**: Selection given by a player. Players take one action at once.
- **Utility**: A value defined for each player's action, an incentive given to each player for an outcome. The larger the value is, the more benefit players receive.

In game theory, some common conditions are provided. Each players are usually assumed to be rational; This means that each player does their best to maximize their own utility. Players are also assumed to be indifferent to other Players' utilities. The conditions and outcomes based on game theory are expected to be applied accurately in e-commerce transactions. Players are assumed to be clearly aware of the range of possible actions and utilities to be provided for each action for both themselves and other players as well. Each player does not know the other players' actions beforehand, however, they are aware of the range of their possible actions and utilities. The basic assumptions that underlie the theory are that players pursue their profits, and take into account their knowledge or expectation of other players' actions. We assume players take actions and have their own utility functions that is determined by a set of all players' actions. An n-player game Γ is denoted by $\Gamma = (\{A_i\}_{i=1}^n, \{u_i\}_{i=1}^n)$. A_i

P_2

P_1		$\acute{A}$	$\acute{B}$	$\acute{C}$
	A	(11,6)	(7,8)	(8,10)
	B	(8,6)	(10,10)	(6,7)
	C	(8,12)	(4,3)	(10,9)

Figure 4: Two player game.

is a set of actions of each player i(P_i from now on). Player P_i selects an action $a_i \in A_i$. u_i is a utility function of P_i. $N = \{P_1, P_2, \dots P_n\}$ is the set of all players. The game is played by having every player takes action $a_i \in A_i$ simultaneously. The payoff to P_i is given by $u_i(\boldsymbol{a})$, where $\boldsymbol{a}$ is the tuple of each player's action ($\boldsymbol{a} = (a_i, \dots, a_n)$). P_i prefers outcome $\boldsymbol{a}$ to outcome $\acute{\boldsymbol{a}}$ iff $u_i(\boldsymbol{a}) \geq u_i(\acute{\boldsymbol{a}})$. We say P_i strictly prefers outcome $\boldsymbol{a}$ to outcome $\acute{\boldsymbol{a}}$ if $u_i(\boldsymbol{\alpha}) > u_i(\acute{\boldsymbol{\alpha}})$ and P_i weakly prefers $\boldsymbol{a}$ to $\acute{\boldsymbol{a}}$ if $u_i(\boldsymbol{a}) \geq u_i(\acute{\boldsymbol{a}})$. We assume that information of all players' possible actions A=$A_1 \times \cdots \times A_n$ and utility functions $u = u_1 \times \cdots \times u_n$ are common knowledge among the players. We show an example of two-player game in Fig. 4. It can be represented in a matrix form by labeling actions of A_1 to rows and A_2 to columns. The entry in the cell at row $a_1 \in A_1$ and column $a_2 \in A_2$ contains a tuple (u_1,u_2) indicating the payoffs to P_1 and P_2, respectively, given the outcome $\boldsymbol{a} = (a_1,a_2)$. The example in Fig. 4 represents a game where $A_1 = \{A, B, C\}$, $A_2 = \{\acute{A}, \acute{B}, \acute{C}\}$, and e.g. $u_1(A, \acute{A}) = 11$ and $u_2(A, \acute{A}) = 6$.

2.6.1 Nash equilibrium

If players play a game and P_1 knows the actions the other players will take, P_1 will select an action $a_1 \in A_1$ that maximizes $u_1(\boldsymbol{a})$. If a_1 is the best way a_1 is called a best response of P_1 to the actions of the other players. If for every player's action a_i is the best response to the other actions, we call the tuple of actions ($\boldsymbol{a} = (a_1, ..., a_n) \in A$) a Nash equilibrium. We define $\boldsymbol{a_{-i}} = (a_1, .., a_{i-1}, a_{i+1}, .., a_n)$ and let $(\acute{a_i}, \boldsymbol{a_{-i}})$ denote $(a_1, ..., a_{i-1}, \acute{a_i}, a_{i+1}, ..., a_n)$.

Definition 1 Let $\Gamma = (A_i, u_i)$ be an n player game. A strategy $\boldsymbol{a}$ is a Nash equilibrium if for all i, it holds that $u_i(\acute{a_i}, \boldsymbol{a_{-i}}) \leq u_i(\boldsymbol{a})$.

In a Nash equilibrium each player can not receive an additional profit by deviating its strategy. In the example in Fig. 4, P_1 may think that P_2 select $\acute{A}$ to receive the maximum payoff 12 ($(a_1,a_2) = (\mathrm{C},\acute{A})$), so P_1 may select strategy A to receive the maximum payoff 11 under the assumption that P_2 will take $\acute{A}$. However, if P_2 thinks that P_1 takes this strategy, $\acute{C}$ becomes a better strategy for P_2.

$u_1(A, \acute{B}) \leq u_1(B, \acute{B}) \geq u_1(C, \acute{B})$

$u_2(B, \acute{A}) \leq u_2(B, \acute{B}) \geq u_2(B, \acute{C})$

So B is the best response to actions of P_2 and $\acute{B}$ is the best response to actions of P_1. In this case, the set of actions $(B, \acute{B})$ fulfills the condition of a Nash equilibrium $u_i(\acute{a_i}, \boldsymbol{a_{-i}}) \leq u_i(\boldsymbol{a})$ for all i.

2.6.2 Mixed strategy Nash equilibrium

The notion of mixed strategy Nash equilibrium is designed to model a steady state of a game in which the players' choices are not deterministic but are regulated by probabilistic rules. We need to add a players' preference over distribution on A to the model of a game. Following the current convention in game theory, we assume that the preference relation of each player i satisfies the assumptions of von Neumann and Morgenstern, so that it can be represented by the expected value of some function u_i:$A \rightarrow R$. We denote by the set of probability distributions over A_i and refer to a member of A_i as a mixed strategy of player i. We assume that the players' mixed strategies are independent randomizations. For clarity, we sometimes refer to member of A_i as a pure strategy.

2.6.3 Correlated equilibrium

The concept of correlated equilibrium is suggested in [**?**]. It may give a better payoff than Nash equilibrium for every player P_i. A correlated equilibrium can be described by means of a joint distribution over the strategy sets.

Let $\Gamma = (\{A_i\}_{i=1}^n, \{u_i\}_{i=1}^n)$ be an n-player game. $\boldsymbol{\alpha} \in A_1 \times \cdots \times A_n$ denotes the set of n-tuple strategies of Γ. We assume the existence of external party M called the mediator and define a mediated version of Γ which relies on M.

The game is now played in two stages: first, the mediator chooses a tuple of actions $\boldsymbol{a} = (a_1, \ldots, a_n) \in A$ according to some known distribution D, and then

P_2

P_1		C	D
	C	(4,4)	(1,5)
	D	(5,1)	(0,0)

Figure 5: An example of "Chicken Game".

P_2

P_1		C	D
	C	1/4	1/4
	D	1/4	1/4

Figure 6: Distribution of mixed Nash equilibrium s^3.

P_2

P_1		C	D
	C	1/3	1/3
	D	1/3	0

Figure 7: Distribution of Correlated equilibrium s^*.

hands the recommendation a_i to player P_i. The players then play Γ as before by choosing any action in their respective action sets. Players are supposed to follow the recommendation of M, and it is the best response for each player to realize a correlated equilibrium. To formally define this notion, let $u_i(\acute{a}_i, \boldsymbol{a_{-i}}|a_i)$ denote the expected utility of P_i, given that it plays action $\acute{a}_i$ after having received recommendation a_i and all other players play their recommended actions $\boldsymbol{a_{-i}}$.

Definition 2 Let $\Gamma = (A_i, u_i)$. A distribution $D \in \Delta(A)$ is a correlated equilibrium if for all $\boldsymbol{a} = (a_1, \ldots, a_n)$ in the support of M, all i, and all $\acute{a}_i \in A_i$, it holds that

$u_i(\acute{a}_i, \boldsymbol{a_{-i}}|a_i) \leq u_i(\boldsymbol{a}|a_i)$.

We consider a simple 2×2 game, the so-called game of "Chicken" shown

in the Figure 5. Here each player can either "dare" (D) or "chicken out" (C). The combination (D,D) has a devastating effect on both players (payoffs [0,0]), (C,C) is quite good (payoffs [4,4]), while each player would ideally prefer to dare while the other chickens-out (giving him 5 and the opponent 1). While the "wisest" pair of actions is (C,C), this is not a Nash equilibrium, since both players are willing to deviate to D (believing that the other player will stay at C). So, Nash equilibria are $s^1 = (D,C)$, $s^2 = (C,D)$ in the pure strategy game. However, if we assume players' strategies are probability distributions, mixed strategy Nash equilibria is seen as: $s^3 = (\frac{1}{2} \cdot D + \frac{1}{2} \cdot C, \frac{1}{2} \cdot D + \frac{1}{2} \cdot C)$. The respective Nash equilibrium payoffs are [5,1], [1,5] and $[\frac{5}{2}, \frac{5}{2}]$. We see that the first two pure strategy Nash equilibria are unfair, while the last mixed equilibrium has small payoffs, since the mutually undesirable outcome (D,D) happens with non-zero probability $\frac{1}{4}$ in the product distribution. The best "fair" strategy profile in the convex hull of the Nash equilibria is the combination $\frac{1}{2}s^1 + \frac{1}{2}s^2 = (\frac{1}{2}(C,D), \frac{1}{2}(D,C))$, yielding payoffs [3,3]. On the other hand, the profile $s^* = (\frac{1}{3}(C,D) + \frac{1}{3}(D,C) + \frac{1}{3}(C,C))$ is a correlated equilibrium, yielding payoffs $[3\frac{1}{3}, 3\frac{1}{3}]$ outside any convex combination of Nash equilibria. To briefly see that this is a correlated equilibrium, consider the "row player" 1 (same works for player 2). If it is recommended to play C, its expected payoff is $\frac{1}{2} \cdot \frac{1}{2} \cdot 1 = \frac{5}{2}$, since assuming the action of the player 1 is C, player 2 is recommended to play C and C with probability $\frac{1}{2}$ each. If player 1 switched to D, its expected payoff would still be $\frac{1}{2} \cdot 5 + \frac{1}{2} \cdot 0 = \frac{5}{2}$, making player 1 reluctant to switch. Similarly, if player 1 is recommended D, it knows that player 2 plays C (as (D,D) is never played in s^*), so its payoff is 5. Since this is the maximum payoff of the game, player 1 would not benefit by switching to C in this case. Thus, we indeed have a correlated equilibrium, where each player's payoff is $\frac{1}{3}(1 + 5 + 4) = 3\frac{1}{3}$, as claimed.

2.6.4 Implementing the mediator

We introduce how to remove the mediator using cryptography. We assume the existence of generic secure two-party protocols and show how to achieve our goal by using such protocols in the game-theoretic and cryptographic setting. In other words, the players remain selfish and rational, even when running the

cryptographic protocol. We give an efficient implementation for cryptographic protocols. To remove the mediator, we assume that

1. The players are computationally bounded.
2. The players can communicate prior to playing the original game.

which seem to be quite natural and minimalistic assumptions. To incorporate communication into the game, we consider an extended game, which is composed of two parts: First the players are given the security parameter and they freely exchange messages (i.e., execute any two-party protocols), then each player locally selects its moves, and finally all players execute their move simultaneously. The payoffs players receive are just the corresponding payoffs of the original game applied to the players' simultaneous moves at the last step. The notions of a strategy and a strategy profile are straightforwardly generalized from those of the basic game, except that they are full-fledged probabilistic algorithms telling each player what to do in each situation. We now define the notion of a computational Nash equilibrium of the extended game, where the strategies of both players are restricted to probabilistic polynomial time (PPT). Also, since we are talking about a computational model, the definition must account for the fact that the players may break the underlying cryptographic scheme with negligible probability (e.g. by guessing the secret key), and gaining some advantage in the game.

Chapter 3 New efficient auction protocol

In this chapter, we show bit-slice auction protocols based on the evaluation of multivariate polynomials of a total degree two on encrypted values. For the first price auction, we compose a secure auction protocol on only 2-DNF formula on encrypted bits. (We do not need to use mix-and-match protocol anymore). On the other hand, for the second price auction, we still need to use the mix-and-match protocols for several times. At first techniques used in this chapter are introduced. Then we propose the three efficient auction protocols for 1st price auction, 2nd price auction and $M+1st$ price auction. After that the security and of proposed protocols is discussed. Finally we compare the efficiency of proposed protocols with previous researches.

3.1 Mix and match protocol

The mix-and-match protocol is a general multiparty protocol proposed by [**?**]. It uses a homomorphic encryption scheme and a MIX net. This model involves n players, denoted by $P_1, P_2, ..., P_n$ and assumes that there exists a public board. The players agree in advance on the presentation of the target function, f as a circuit C_f. The aim of the protocol is for players to compute $f(B_1, ..., B_n)$ without revealing any additional information. Its outline is as follows.

1. **Input stage:** Each $P_i (1 \leq i \leq n)$ computes ciphertexts of the bits of B_i and broadcasts them. She proves that each ciphertext represents 0 or 1 by using the zero-knowledge proof technique in [**?**].

2. **Mix and Match stage:** The players blindly evaluate each gate, G_j in order.

3. **Output stage:** After evaluating the last gate G_N, the players obtain O_N, a ciphertext encrypting $f(B_1, ..., B_n)$. They jointly decrypt this ciphertext value to reveal the output of the function f.

3.1.1 Requirements for the encryption function

Let E be a public-key probabilistic encryption function. We denote by $E(m)$ the set of encryptions for a plaintext m and by $c \in E(m)$ a particular encryption of m.

Function E must satisfy the following properties.

Homomorphic property There exists a polynomial time computable operations, $^{-1}$ and $\otimes$, as follows for a large prime q.

1.If $c \in E(m)$, then $c^{-1} \in E(-m \bmod q)$.

2.If $c_1 \in E(m_1)$ and $c_2 \in E(m_2)$, then $c_1 \otimes c_2 \in E(m_1 + m_2 \bmod q)$.

For a positive integer a, define

$$a \cdot e = \underbrace{c \otimes c \otimes \cdots \otimes c}_{a}$$

Random re-encryption Given $c \in E(m)$, there is a probabilistic re-encryption algorithm that outputs $c' \in E(m)$, where c' is uniformly distributed over $E(m)$.

Threshold decryption For a given ciphertext $c \in E(m)$, any t out of n players can decrypt c along with a zero-knowledge proof of the correctness. However, any t-1 out of n players cannot decrypt c.

Such $E(\cdot)$ can be obtained by slightly modifying the ElGamal encryption scheme over a group G of order $|G| = q$. The secret key x is a random element $x \in \mathbb{Z}_q^*$ and the public key is $y = g^x$. And an encryption of m is given by

$$(g^r, g^m y^r) \in E(m),$$

where $r \in \mathbb{Z}_q^*$ is a random element. For ciphertexts, $^{-1}$ and $\otimes$ are defined as

$$(u_1, v_1)^{-1} = (u_1^{-1}, v_1^{-1}) \text{ and}$$
$$(u_1, v_1) \otimes (u_2, v_2) = (u_1 u_2, v_1 v_2), \text{ respectively.}$$

Then it is easy to see that the homomorphic property is satisfied. A re-encryption of $(u, v) \in E(m)$ is given by $(u', v') = (g^{r'}u, y^{r'}v)$ for random element $r' \in \mathbb{Z}_q^*$. In the threshold type of ElGamal encryption, each player has a share of secret key $x_i (i = 1, 2, ..., m)$ and publishes a share of public key $y_i = g^{x_i}$. Each player needs to broadcast $O(1)$ message and compute $O(n)$ exponentiations in the threshold decryption.

3.1.2 MIX protocol

A MIX protocol (proposed in [**?**]) takes a list of ciphertexts, $(\xi_1, ..., \xi_L)$ and outputs a permuted and re-encrypted list of the ciphertexts $(\xi'_1, ..., \xi'_L)$ without revealing the relationship between $(\xi_1, ..., \xi_L)$ and $(\xi'_1, ..., \xi'_L)$, where ξ_i or ξ'_i can be a single ciphertext c, or a list of l ciphertexts, $(c_1, ..., c_l)$, for some $l > 1$. For all players to verity the validity of $(\xi'_1, ..., \xi'_L)$, we use the universal verifiable MIX net protocol suggested by [**?**].

3.1.3 Plaintext equality test

Given two ciphertexts $c_1 \in E(v_1)$ and $c_2 \in E(v_2)$, this protocol checks if $v_1 = v_2$. Let $c_0 = c_1 \otimes c_2^{-1}$.

(Step 1) For each player P_i (where i = 1,...,m):

P_i chooses a random element $a_i \in \mathbb{Z}_q^*$ and computes $z_i = a_i \cdot c_0$. He broadcasts z_i and proves the validity of z_i in zero-knowledge.

(Step 2) Let $z = z_1 \otimes z_2 \otimes \cdots \otimes z_n$. The players jointly decrypt z using threshold verifiable decryption and obtain plaintext v. Then it holds that

$$v = \begin{cases} 0 & if\ v_1 = v_2 \\ random & otherwise \end{cases}$$

3.1.4 Mix and match stage

For each logical gate, $G(x_1, x_2)$ of a given circuit, n players jointly computes $E(G(x_1, x_2))$ from $c_1 \in E(x_1)$ and $c_2 \in E(x_2)$ keeping x_1 and x_2 secret. For simplicity, we show the mix-and-match stage for AND gate.

1. n players first consider the standard encryption of each entry of table shown below.

2. By applying a MIX protocol to the four rows of the table, n players jointly

Table 3: Mix-and-match table for AND.

x_1	x_2	$x_1 \wedge x_2$
$a'_1 \in E(0)$	$b'_1 \in E(0)$	$c'_1 \in E(0)$
$a'_2 \in E(0)$	$b'_2 \in E(1)$	$c'_2 \in E(0)$
$a'_3 \in E(1)$	$b'_3 \in E(0)$	$c'_3 \in E(0)$
$a'_4 \in E(1)$	$b'_4 \in E(1)$	$c'_4 \in E(1)$

compute blinded and permuted rows of the table. Let the ith row be (a'_i, b'_i, c'_i) for i = 1,...,4.

3. n players next jointly find the row i such that the plaintext of c_1 is equal to that of a'_i and the plaintext of c_2 is equal to that of b'_i by using the plaintext equality test protocol.

4. For the row i, it holds that $c'_i \in E(x_1 \wedge x_2)$.

3.2 Bit-slice auction circuit

We introduce an efficient auction circuit called the bit-slice auction circuit suggested by [**?**].

Suppose that $B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$ is the highest bidding price and a bid of a player i is $B_i = (b_i^{(k-1)}, ..., b_i^{(0)})_2$, where $()_2$ is the binary expression. Then the proposed circuit first determines $b_{max}^{(k-1)}$ by evaluating the most significant bits of all the bids. It next determines $b_{max}^{(k-2)}$ by looking at the second most significant bits of all the bids, and so on.

For two m-dimensional binary vectors $\mathbf{X} = (x_1, ..., x_m)$ and $\mathbf{Y} = (y_1, ..., y_m)$,

$$\mathbf{X} \wedge \mathbf{Y} = (x_1 \wedge y_1, ..., x_m \wedge y_m)$$

Let D_j be the highest price when considering the upper j bits of the bids. That is,

$$D_1 = (b_{max}^{(k-1)}, 0, ..., 0)_2$$
$$D_2 = (b_{max}^{(k-1)}, b_{max}^{(k-2)}, 0, ..., 0)_2$$
$$\cdots$$

$$D_k = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$$

In the j-th round, we find $b_{max}^{(k-j)}$ and eliminate a player P_i such that his bid satisfies $B_i < D_j$. For example, in the case of $j = 1$, a player i is eliminated if his bid $B_i < D_1$. By repeating this operation for 1 to $k-1$, at the end the remaining bidder is the winner.

For this purpose, we update $\mathbf{W} = (w_1, ..., w_m)$ such that

$$w_i = \begin{cases} 1 & if\ B_i \geq D_j \\ 0 & otherwise \end{cases}$$

for $j = 1$ to k. The circuit is obtained by implementing the following algorithm. For given m bids, $B_1, ..., B_m$, V_j is defined as

$$V_i = (b_1^{(j)}, ..., b_m^{(j)})$$

for $j = 0$,...,$k-1$, that is, V_j is the vector consisting of the $(j+1)$th lowest bit of each bid. Let $\mathbf{W} = (w_1, ..., w_m)$, where each $w_j = 1$. For $j = k-1$ to 0, perform the following;

(Step 1) For $\mathbf{W} = (w_1, ..., w_m)$, let

$$S_j = \mathbf{W} \wedge V_j$$
$$= (w_1 \wedge b_1^{(j)}, ..., w_m \wedge b_m^{(j)})$$
$$b_{max}^{(j)} = (w_1 \wedge b_1^{(j)}) \vee \cdots \vee (w_m \wedge b_m^{(j)}) \ .$$

(Step 2) If $b_{max}^{(j)} = 1$, then let $\mathbf{W} = S_j$.

Then the highest price is obtained as $B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$. Let the final $\mathbf{W}$ be $(w_1, ..., w_m)$. Then P_i is the winner if and only if $w_i = 1$. We summarize the algorithm as the following theorem.

Theorem 1 [**?**] In the bit-slice auction above,

- B_{max} is the highest bidding price.
- For the final $\mathbf{W} = (w_1, ..., w_m)$, P_i is a winner if and only if $w_i = 1$ and P_i is the only player who bids the highest price B_{max}.

3.3 Evaluating 2-DNF formulas on ciphertexts

Given encrypted Boolean variables $x_1, ..., x_n \in \{0, 1\}$, a mechanism for public evaluation of a 2-DNF formula was suggested in [**?**]. They presented a homomorphic public key encryption scheme based on finite groups of composite order that supports a bilinear map. In addition, the bilinear map allows for one multiplication on encrypted values. As a result, their system supports arbitrary additions and one multiplication on encrypted data. This property in turn allows the evaluation of multivariate polynomials of a total degree of two on encrypted values.

3.3.1 Bilinear groups

Their construction makes use of certain finite groups of composite order that supports a bilinear map. We use the following notation.

1. $\mathbb{G}$ and $\mathbb{G}_1$ are two (multiplicative) cyclic groups of finite order n.
2. g is a generator of $\mathbb{G}$.
3. e is a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$.

3.3.2 Subgroup decision assumption

We define algorithm $\mathcal{G}$ such that given security parameter $\tau \in \mathbb{Z}^+$ outputs a tuple

$(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$ where $\mathbb{G}, \mathbb{G}_1$ are groups of order $n = q_1 q_2$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ is a bilinear map. On input τ, algorithm $\mathcal{G}$ works as indicated below,

1. Generate two random primes, q_1, q_2 and set $n = q_1 q_2 \in \mathbb{Z}$.
2. Generate a bilinear group $\mathbb{G}$ of order n as described above. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be the bilinear map.
3. Output $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$.

   We note that the group action in $\mathbb{G}$ and $\mathbb{G}_1$ as well as the bilinear map can be computed in polynomial time.

Let $\tau \in \mathbb{Z}^+$ and let $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$ be a tuple produced by $\mathcal{G}$ where $n = q_1 q_2$. Consider the following problem. Given $(n, \mathbb{G}, \mathbb{G}_1, e)$ and an element $x \in \mathbb{G}$, output '1' if the order of x is q_1 and output '0' otherwise, that is, without knowing the factorization of the group order n, decide if an element x is in a subgroup of $\mathbb{G}$. We refer to this problem as the subgroup decision problem.

3.3.3 Homomorphic public key system

We now describe the proposed public key system which resembles the Paillier [**?**] and the Okamoto-Uchiyama encryption schemes [**?**]. We describe the three algorithms comprising the system.

KeyGen Given a security parameter $\tau \in \mathbb{Z}$, run $\mathcal{G}$ to obtain a tuple $(q_1, q_2, \mathbb{G}, \mathbb{G}_1, e)$. Let $n = q_1 q_2$. Select two random generators, g and $u \xleftarrow{R} \mathbb{G}$ and set $h = u^{q_2}$. Then h is a random generator of the subgroup of $\mathbb{G}$ of order q_1. The public key is $PK = (n, \mathbb{G}, \mathbb{G}_1, e, g, h)$. The private key is $SK = q_1$.

Encrypt(PK, M) We assume that the message space consists of integers in set $\{0, 1, ..., T\}$ with $T < q_2$. We encrypt binary representation of bids in our main application, in the case $T = 1$. To encrypt a message m using public key PK, select a random number $r \in \{0, 1, ..., n-1\}$ and compute

$$C = g^m h^r \in \mathbb{G}.$$

Output C as the ciphertext.

Decrypt(SK, C) To decrypt a ciphertext C using the private key $SK = q_1$, observe that $C^{q_1} = (g^m h^r)^{q_1} = (g^{q_1})^m$ Let $\hat{g} = g^{q_1}$. To recover m, it suffices to compute the discrete log of C^{q_1} base $\hat{g}$.

3.3.4 Homomorphic properties

The system is clearly additively homomorphic. Let $(n, \mathbb{G}, \mathbb{G}_1, e, g, h)$ be a public key. Given encryptions C_1 and $C_2 \in \mathbb{G}_1$ of messages v_1 and $v_2 \in \{0, 1, ..., T\}$ respectively, anyone can create a uniformly distributed encryption of $v_1 + v_2$ mod n by computing the product $C = C_1 C_2 h^r$ for a random number $r \in \{1, ..., n-1\}$. More importantly, anyone can multiply two encrypted messages once using the bilinear map. Set $g_1 = e(g, g)$ and $h_1 = e(g, h)$. Then g_1 is of order n and h_1 is of order q_1, also $h = g^{\alpha q_2}$ for some (unknown)$\alpha \in \mathbb{Z}$. Suppose we are given two ciphertexts $C_1 = g^{v_1} h^{r_1} \in \mathbb{G}$ and $C_2 = g^{v_2} h^{r_2} \in \mathbb{G}$. To build an encryption of product $v_1 \cdot v_2$ mod n given only C_1 and C_2, 1) select random $r \in \mathbb{Z}_n^*$, and 2) set $C = e(C_1, C_2) h_1^r \in \mathbb{G}_1$. Then

$$C = e(C_1, C_2) h_1^r = e(g^{v_1} h^{r_1}, g^{v_2} h^{r_2}) h_1^r$$
$$= g_1^{v_1 v_2} h_1^{v_1 r_2 + v_2 r_1 + q_2 r_1 r_2 \alpha + r} = g_1^{v_1 v_2} h_1^{r'} \in \mathbb{G}_1$$

where $r' = v_1 r_2 + v_2 r_1 + q_2 r_1 r_2 \alpha + r$ is distributed uniformly in $\mathbb{Z}_n$ as required. Thus, C is a uniformly distributed encryption of $v_1 v_2$ mod n, but in the group $\mathbb{G}_1$ rather than $\mathbb{G}$ (this is why we allow for just one multiplication). We note that the system is still additively homomorphic in $\mathbb{G}_1$. For simplicity, in this thesis we denote an encryption of message v in $\mathbb{G}$ as $E_G(v)$ and one in $\mathbb{G}_1$ as $E_{G_1}(v)$.

3.4 First price auction

We assume n players, $P_1, ..., P_n$ and a set of auction managers, AM. The players bid their encrypted prices, and through the protocol they publish encrypted flags whether they are still in the auction. AM jointly decrypts the result of the protocol. Players find the highest price through the protocol and the winner by decrypting the result.

3.4.1 Setting

AM jointly generates and shares private keys among auction managers using the technique described in [**?**].

3.4.2 Bidding Phase

Each player P_i computes a ciphertext of his bidding price, B_i as

$$ENC_i = (c_{i,k-1}, ..., c_{i,0})$$

where $c_{i,j} \in E_G(b_i^{(j)})$, and publishes ENC_i on the bulletin board. He also proves in zero-knowledge that $b_i^{(j)} = 0$ or 1 by using the technique described in [**?**].

3.4.3 Opening Phase

Suppose that $c_1 = g^{b_1} h^{r_1} \in E_G(b_1)$ and $c_2 = g^{b_2} h^{r_2} \in E_G(b_2)$, where b_1, b_2 are binary and $r_1, r_2 \in \mathbb{Z}_n^*$ are random numbers. We define two polynomial time computable operations Mul and $\otimes$ by applying a 2DNF formula for AND, OR respectively.

$$Mul(c_1, c_2) = e(c_1, c_2) = e(g^{b_1} h^{r_1}, g^{b2} h^{r_2}) \in E_{G_1}(b_1 \wedge b_2)$$
$$c_1 \otimes c_2 = g^{b_1} h^{r_1} \cdot g^{b2} h^{r_2} = g^{b_1 + b_2} h^{r_1 + r_2} \in E_G(b_1 + b_2)$$

by applying a 2DNF formula for AND.

AM generates $W = (w_1, ..., w_m)$, where each w_j =1, and encrypts them as $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$. AM shows that $\widetilde{W}$ is the encryption of (1,...,1) with the verification protocols.

(Step 1) For $j = k$ -1 to 0, perform the following.

(Step 1-a) For $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM computes $s_{i,j} = Mul(\tilde{w}_i, c_{i,j})$ for each player i, and

$$S_j = (Mul(\tilde{w}_1, c_{1,j}), ..., Mul(\tilde{w}_m, c_{m,j}))$$
$$h_j = Mul(\tilde{w}_1, c_{1,j}) \otimes \cdots \otimes Mul(\tilde{w}_m, c_{m,j})$$

(Step 1-b) AM takes a plaintext equality test regarding whether h_j is an encryption of 0. If h_j is an encryption of 0, AM publishes 0 as the value of $b_{max}^{(j)}$ and proves it with the verification protocols, otherwise, AM publishes 1 as the value of $b_{max}^{(j)}$.

(Step 1-c) If $b_{max}^{(j)} = 1$, then each player creates a new encryption $\tilde{w}_i$ which has the same plaintext value of $s_{i,j}$, otherwise he uses w_i for the next bit. And the player shows the validity of computation with zero-knowledge proof.

(Step 2) For the final $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM decrypts each $\tilde{w}_i$ with the verification protocols and obtains plaintext w_i.

The highest price is obtained as

$B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$. P_i is a winner if and only if $w_i = 1$.

3.5 Second price auction

In the second price auction, the information that players can find is the second highest price and the bidder of the highest price. To maintain secrecy of the highest bid through the protocol, we need to use the mix-and-match protocol. However, we can reduce the number of times we use it. As a result, the proposed protocol is more efficient than that in [**?**]. Here, we define three types of new tables, $Select_m$, MAP_1 and MAP_2 for the second price auction. In this protocol, MAP_1 and MAP_2 tables are created among AM before an auction, on the other hand $Select_m$ is created through the protocol corresponding to players' inputs. AM computes jointly for distributed decryption of plaintext equality test. Table $Select_m$ is also used for the second price auction protocol in [**?**]; MAP_1 and MAP_2 are new tables that we propose. Given a message m, MAP_1 and MAP_2 are tables for mapping a encrypted value $a_1 \in E_{G_1}(m)$ (which is an output of a computation with one multiplication) to $a_2 \in E_G(m)$.

Table $Select_m$ has $2k + 1$ input bits and k output bits as follows.

Table 4: Table for MAP_1

x_1	x_2
$a_1 \in E_{G_1}(0)$	$b_1 \in E_G(0)$
$a_2 \in E_{G_1}(1)$	$b_2 \in E_G(1)$

Table 5: Table for MAP_2

x_1	x_2
$a_1 \in E_{G_1}(0)$	$b_1 \in E_G(0)$
$a_2 \in E_{G_1}(1)$	$b_2 \in E_G(1)$
$\cdots$	$b_i \in E_G(1)$
$a_{m+1} \in E_{G_1}(m)$	$b_{m+1} \in E_G(1)$

$$Select_m(b, x^{(m-1)}, ..., x^{(0)}, y^{(m-1)}, ..., y^{(0)}) = \begin{cases} (x^{(m-1)}, ..., x^{(0)}) & if\ b = 1 \\ (y^{(m-1)}, ..., y^{(0)}) & otherwise \end{cases}$$

For two encrypted input vectors $(x^{(k-1)}, ..., x^{(0)})$ and $(y^{(k-1)}, ..., y^{(0)})$, b is an encrytption of check bit that selects which vector to output,
$(x^{(k-1)}, ..., x^{(0)})$ or $(y^{(k-1)}, ..., y^{(0)})$. For secure computation, AM re-encrypts an output vector. In this protocol, $Select_m$ table is created through the auction to update W corresponding to an input value $E(b_j)$.
The function of table MAP_1 is a mapping
$x_1 \in \{E_{G_1}(0), E_{G_1}(1)\} \rightarrow x_2 \in \{E_G(0), E_G(1)\}$.

The table MAP_2 is the one for mapping
$x_1 \in \{E_{G_1}(0), E_{G_1}(1), ..., E_{G_1}(m)\} \rightarrow x_2 \in \{E_G(0), E_G(1)\}$.

These tables can be composed on using the mix-and-match protocol because the Boneh-Goh-Nissim encryption has homomorphic properties. The setting and bidding phases are the same as the first price auction, so we start from the opening phase.

3.5.1 Opening phase

Let $\widetilde{W} = (\tilde{w}_1, .., \tilde{w}_m)$, where each $\tilde{w}_j \in E_G(1)$ shown above.

(Step 1) For $j = k$ -1 to 0, perform the following.

(Step 1-a) For $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM computes $s_{i,j} = Mul(\tilde{w}_i, e_{i,j})$ for each player i, and

$$S_j = (Mul(\tilde{w}_1, c_{1,j}), ..., Mul(\tilde{w}_m, c_{m,j}))$$
$$h_j = Mul(\tilde{w}_1, c_{1,j}) \otimes \cdots \otimes Mul(\tilde{w}_m, c_{m,j})$$

(Step 1-b) AM uses table MAP_1 for $s_{i,j}$ for each i and find the values of $\tilde{s}_{i,j}$. Let $\widetilde{S}_j = (\tilde{s}_{1,j}, ..., \tilde{s}_{m,j})$. AM also uses the table MAP_2 for h_j as an input value. By using this table, AM retrieve $E(b_j) \in E_G(0)$ if h_j is a ciphertext of 1, otherwise he retrieves $E(b_j) \in E_G(1)$.

(Step 1-c) AM creates the table $Select_m$ as input values $(E(b_j), \widetilde{S}_j, \widetilde{W})$.
By using table $Select_m$, if $E(b_j)$ is the encryption of 1, AM updates $\widetilde{W} = \widetilde{S}_j$, otherwise $\widetilde{W}$ remains unchanged.

(Step 2) For the final $\widetilde{W} = (\tilde{w}_1, ..., \tilde{w}_m)$, AM decrypts each $\tilde{w}_i$ with verification protocols and obtains the plaintext w_i. P_i is a winner if and only if $w_i = 1$. We remove the player who bids the highest price and run the first price auction protocol again. The second highest price is obtained
as $B_{max} = (b_{max}^{(k-1)}, ..., b_{max}^{(0)})_2$.

Verification protocols

Verification protocols are the protocols for players to confirm that AM decrypts the ciphertext correctly. By using the protocols, each player can verify the result of the auction is correct. Denote b is a plaintext and C is a BGN encryption of b ($C = g^b h^r$), where g, h, r are elements used in BGN scheme and $f = (h)(g^b)^{-1}$. Before a player verifies whether b is the plaintext of C, the player has to prove that a challenge ciphertext $C' = g^x f^r$ is created by himself with zero-knowledge proof that he has the value of x.

1. A player proves that he has random element $x \in Z_n^*$ with zero-knowledge proof.
2. The player computes $f = (h)(g^b)^{-1}$ from the published values, h, g and b, and select a random integer $r \in \mathbb{Z}_n^*$.He sends $C' = g^x f^r$ to AM.

3. AM decrypts C and sends value x' to the player.
4. The player verifies whether $x = x'$. AM can decrypt C' correctly only if order$(f) = q - 1$, which means AM correctly decrypts C and publishes b as the plaintext of C'.

3.6 $M + 1$st price auction

In this section, we show an efficient $M + 1st$ price auction based on bit-slice auction protocols. Compared to previous works on secure $M+1st$ price auctions, the proposed protocol is more efficient because bidding prices are represented as binary numbers. However if a quite large number of players participate in an auction, it still needs high computation costs, because the complexity of proposed protocol is a polynomial of m for the m-player auction. If some players bid the same price which is more than M highest price, such as a case 2 players bid the same price as 3rd highest price for 5-player auction for 3 goods, this protocol does not work well. (Regarding to this situation called Tie-Break, see [**?**] for more details.) At the end of auction, winners and winning price can not be decided. We show how to find the winners and the winning bidding price with unencrypted bidding prices. Through an auction, players are labeled as three types of players' statuses, winner(s), candidate(s) and survivor(s) described as follow.

- $Winner$: a player who decided to be a winner.
- $Candidate$: a player who is not decided to be a winner but has a possibility of $M + 1st$ highest bidder.
- $Survivor$: a candidate on the current and his bid on the bit is 1.

This auction protocol starts from the highest bit of players' bidding prices and proceeds to lower one bit by one bit. At the beginning of the auction, all players are Candidates since no player is decided as a Winner and all players have possibilities to win the auction. On each bit, a status of a player is decided by comparing players' bidding prices. If a player's bidding price is found to be larger than $M + 1st$ highest bit, his status becomes a Winner. On the other hand, the bidding prices is found to be smaller than $M + 1st$ highest bit, he loses a status of a Candidate, because he no longer has a possibility to win

the auction. Otherwise, while he has a chance to be a Winner or $M+1st$ highest bidder, he keeps his status a Candidate. At the end of the auction, the winners and the winning price is found according to the players' bidding prices. To explain precisely, we also define the players in the variables of winner(s), candidate(s) and survivor(s) on j-th bid as W_j, C_j and S_j respectively and the numbers of elements whose value is 1 in W_j and S_j as $|W_j|$ and $|S_j|$.

- $W_j[1 \dots m]$: $W_j[i]$=1 if player P_i is decided to be a winner by upper $k-j$ bits of the bid.
- $C_j[1 \dots m]$: $C_j[i]$=1 if player P_i is not decided to be a winner but has a possibility of $M+1st$ highest bidder by upper $k-j$ bits of the bid.
- $S_j[1 \dots m]$: $S_j[i]$=1 if player P_i is a candidate on j-th bit ($C_j[i]$=1) and his bid on j-th bit is 1.

Suppose that $Z_i = (z_i^{(k-1)}, \dots, z_i^{(0)})_2$ is the bid of player i, and $Z_{M+1st} = (z_{M+1st}^{(k-1)}, \dots, z_{M+1st}^{(0)})_2$ is the $M+1st$ highest bidding price where $()_2$ is the binary expression. The winners and winning price are found by the following protocol.

As initial setting, we set $W_k[i] = 0$ $(1 \le i \le m)$ and $C_k[i] = 1$ ($1 \le i \le m$).

For $j = k-1$ to 0

$S_j[i] = C_{j+1}[i] \wedge z_i^{(j)}$ $(1 \le i \le m)$

if $|W_{j+1}| + |S_j| \ge M+1$ then

$W_j[i] = W_{j+1}[i]$ $(1 \le i \le m)$

$C_j[i] = S_j[i]$ ($1 \le i \le m$)

$z_{M+1st}^{(j)} = 1$

else

$W_j[i] = W_{j+1}[i] \vee S_j[i]$ $(1 \le i \le m)$

$C_j[i] = C_{j+1}[i] \wedge \overline{S_j[i]}$ $(1 \le i \le m)$

$z_{M+1st}^{(j)} = 0$

end

end

For a player P_i $(1 \le i \le m)$, if P_i is decided to be a winner by j-th bit from high-order bits of the bid, then $W_j[i] = 1$. If player P_i is not decided to be a winner but has a possibility of $M+1st$ highest bidder on the j-th bit, then $C_j[i]$=1. If $C_j[i]$=1 and j-th bit of P_i's bid is 1, then $S_j[i] = 1$.

Table 6: Example of 5-player auction for 3 goods.

	$j=4$		$j=3$				$j=2$				$j=1$				$j=0$			
	C_4	W_4	K_3	S_3	C_3	W_3	K_2	S_2	C_2	W_2	K_1	S_1	C_1	W_1	K_0	S_0	C_0	W_0
$Z_1 = (1011)_2$	1	0	1	1	0	1	0	0	0	1	1	0	0	1	1	0	0	1
$Z_2 = (0111)_2$	1	0	0	0	1	0	1	1	1	0	1	1	0	1	1	0	0	1
$Z_3 = (0101)_2$	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	1
$Z_4 = (0100)_2$	1	0	0	0	1	0	1	1	1	0	0	0	1	0	0	0	1	0
$Z_5 = (0001)_2$	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
$\|W\|$ and $\|S\|$		0		1		1		3		1		1		2		1		3
Z_{M+1st}				0				1				0				0		

If the number of Winners on $(j+1)$-th bit and Survivors on j-th bit is more than or equal to $M+1$, we keep Winners remained and update players' status Candidates to eliminate players i whose bidding prices are 0 on this bit. If the number of Winners on $(j+1)$-th bit and Survivors on j-th bit is less than $M+1$, Survivors on j-th bit are determined as Winners, so we update W_j as $W_{j+1}[i] \vee S_j[i]$ and eliminate player i that satisfies $S_j[i]$=1.

Theorem 2 In the above algorithm,

- For the vector W_0, P_i is the winner of the auction if and only if $W_0[i] = 1$.
- Z_{M+1st} is the M+1st bidding price.

Proof. We show the values of Winners, Candidates and Survivors satisfy the definition for all l bits by induction and the winning price, Z_{M+1st}, is consistent with the bidding prices of players.
We show that the variables satisfy the definitions through the proposed auction protocol by induction. In this proof, we denote the $M+1st$ bidding players by P_{M+1st}.

- Initial Step:

When $l = k$, following the initial setting, Winner is a null vector, and the statuses of all players are Candidate. $z^{(1)}_{M+1st}$ is a blank(not defined). Thus this situation satisfies the definition of the players statuses.

- Inductive step:

When $l = j+1$ we assume the definition of each player status holds on $(j+1)$-th and upper bits, then we show that the definition of each player status

holds when $l = j$ that is;

(1). If the number of Winners by the upper $(j+1)$-th bits of Z_i and Survivors on j-th bit is more than or equal to $M+1(|W_{j+1}|+|S_j| \geq M+1)$, new Winners can not selected on this bit, because if Survivors become Winners, the number of Winners exceeds the number of goods M. The players in the status of Winners do not change. Survivors (Candidates whose bids on this bit are 1) become Candidates of next bit because they have a chance to be a Winner or P_{M+1st}. The rest of Candidates$(C_{j+1} - S_j)$ lose the auction since their bidding prices are found to be smaller than Z_{M+1st}. Thus, the definition of players' status holds. In this case, Z_{M+1st} is bigger than or equal to the lowest bid of Survivors, which is the $(|W_{j+1}| + |S_j|)$-th highest bid, then P_{M+1st} is categorized as a Survivor. Thus $Z^{(j)}_{M+1st}$ is 1.

(2). If the number of Winners by the upper $(j+1)$-th bits and Survivors on j-th bit is is less than $M+1(|W_{j+1}|+|S_j| < M+1)$, Survivors are decided to be Winners, since their bidding prices are found to be larger than Z_{M+1st}. On the other hand, the players in $C_{j+1} - S_j$ become Candidates, since they still have a chance to be a Winner or P_{M+1st}. Thereby showing that in the both situation the definition of each player status holds when $l = j$. In this case, Z_{M+1st} is smaller than the $(|W_{j+1}| + |S_j|)$-th highest bid and P_{M+1st} is in the group of $C_{j+1} - S_j$. Thus, $Z^{(j)}_{M+1st}$ is 0.

□

We show an example of 5-player auction for 3 goods ($M = 3$) in Table 6. The information we need to find is the first, second and third highest bidders as the winners of the auction and the forth highest bidding price as the winning price. Assume each player's bid as follows,

$$
\begin{aligned}
Z_1 &= (1011)_2 = 11 \\
Z_2 &= (0111)_2 = \ 7 \\
Z_3 &= (0101)_2 = \ 5 \\
Z_4 &= (0100)_2 = \ 4 \\
Z_5 &= (0001)_2 = \ 1
\end{aligned}
$$

So, the winners are P_1, P_2 and P_3 and the winning price is $Z_4 = (0100)_2 = 4$. In Table 6, we denote by K_j the vector comprising the $k - j$-th MSB of each player's bid.

For initial setting $j = 4$, all players are Candidates, since all players have possibilities to win the auction according to the definition of the player status. They are not decided to win the auction yet, so none of players' statuses is Winners.

Next step $j = 3$, only z_1^4 is 1, so P_1 is decided to be Survivor and the number of Winner on upper bit and Survivor on 4th bid is 1. Then, by following the protocol, P_1 becomes Winner and is removed from Candidate. The other players are kept to be Candidates to compete the auction. Next step $j = 2$, bids of Z_2, Z_3 and Z_4 are 1, so they are decided as Survivors. The number of Winner on upper bit and Survivor on 3rd bid is 4, which means P_2, P_3 and P_4 can not decided to be Winners but kept to be Candidates and P_5 already loses the auction. Following the protocol, from the 1st bits of the bids P_1, P_2 and P_3 are decided to be Winners. The winning price $Z_4 = (0100)_2$ is shown in the row of Z_{M+1st} in the Table 6.

We assume m players, $P_1, \ldots, P_m$ and a set of auction managers, AM. The players bid their encrypted prices and broadcast them. The AM runs an auction protocol with the encrypted bids and after the auction AM jointly decrypts the results of the protocol and broadcast it to the players. Players can verify the winning price (the $M + 1st$ price) and the winners from the encrypted bidding prices by using verification protocols. To maintain secrecy of the players' bidding prices through the protocol, we need to use the mix-and-match protocol. We continue to use MAP_1 from previous subsection. Here, we define new table, MAP_3. In the proposed protocol, the MAP_1 and MAP_3 tables are created among AM before an auction. The AM jointly computes values in the mix-and-match table for distributed decryption of plaintext equality test. The function of table MAP_1 is used for transferring encrypted values of 0 and 1 in $\mathbb{G}_1$ to encrypted values of 0 and 1 in $\mathbb{G}$ respectively. This mapping, $x_1 \in \{E_{G_1}(0), E_{G_1}(1)\} \mapsto x_2 \in \{E_G(0), E_G(1)\}$, is shown in Table 4. The table

Table 7: Table for MAP_3

x_1	x_2
$a_1 \in E_{G_1}(0)$	$b_1 \in E_G(0)$
$a_2 \in E_{G_1}(1)$	$b_2 \in E_G(0)$
$\cdots$	$b_i \in E_G(0)$
$a_{M+1} \in E_{G_1}(M)$	$b_{M+1} \in E_G(0)$
$a_{M+2} \in E_{G_1}(M+1)$	$b_{M+2} \in E_G(1)$
$\cdots$	$b_i \in E_G(1)$
$a_{m+1} \in E_{G_1}(m)$	$b_{m+1} \in E_G(1)$

MAP_3 is a function for mapping $x_1 \in \{E_{G_1}(0), E_{G_1}(1), \ldots, E_{G_1}(m)\} \mapsto x_2 \in \{E_G(0), E_G(1)\}$. This is used for transferring encrypted values of {0,...,M} and $M+1$,...,m} in $\mathbb{G}_1$ to encrypted values of 0 and 1 in $\mathbb{G}$, respectively as described in Table 7. These tables can be constructed using the mix-and-match protocol because the Boneh-Goh-Nissim encryption has homomorphic properties.

3.6.1 Setting

AM jointly generates and shares private keys among themselves using the technique described in [**?**].

3.6.2 Bidding Phase

Suppose that a bid of a player i is $Z_i = (z_i^{(k-1)}, \ldots, z_i^{(0)})_2$ and $Z_{M+1st} = (z_{M+1st}^{(k-1)}, \ldots, z_{M+1st}^{(0)})_2$ is the $M+1st$ highest bidding price, where $()_2$ is the binary expression. Each player P_i computes a ciphertext of his bidding price, Z_i, as

$$ENC_i = (b_i^{k-1}, \ldots, b_i^0)$$

where $b_i^j \in E_G(z_i^{(j)})$, and publishes ENC_i on the bulletin board. He also proves in zero-knowledge that $z_i^{(j)} = 0$ or 1 by using the technique described in [**?**].

3.6.3 Opening phase

Suppose that $c_1 = g^{b_1} h^{r_1} \in E_G(b_1)$ and $c_2 = g^{b_2} h^{r_2} \in E_G(b_2)$, where b_1, b_2 are binary, $r_1, r_2 \in \mathbb{Z}_n^*$ are random numbers and $c_1' \in E_{G_1}(b_1)$ and $c_2' \in E_{G_1}(b_2)$. We define two polynomial time computable operations Mul by applying a 2-DNF

formula for AND, and $\otimes$ by the operation of addition.

$$Mul(c_1, c_2) = e(c_1, c_2) = e(g^{b_1}h^{r_1}, g^{b_2}h^{r_2}) \in E_{G_1}(b_1 \wedge b_2)$$
$$c'_1 \otimes c'_2 \in E_{G_1}(b_1 + b_2)$$

AM executes PET for MAP_1 and MAP_3 in this open phase to keep the secrecy of players bidding prices through the auction. Let $C_k = (c_1^k, \ldots, c_m^k)$, where each $c_i^k \in E_G(1)$ and $W_k = (w_1^k, \ldots, w_m^k)$, where each $w_i^k \in E_{G_1}(0)$.

(Step 1) For $j = k$ -1 to 0, perform the following.

(Step 1-a) For $C_j = (c_1^j, \ldots, c_m^j)$, AM computes $s_i^j = Mul(c_i^{j+1}, b_i^j)$ for each player i, and

$$S_j = (s_1^j, \ldots, s_m^j) = (Mul(c_1^{j+1}, b_1^j), \ldots, Mul(c_m^{j+1}, b_m^j))$$
$$h_j = Mul(c_1^{j+1}, b_1^j) \otimes \cdots \otimes Mul(c_m^{j+1}, b_m^j)$$
$$d_j = w_1^j \otimes \cdots \otimes w_m^j$$

(Step 1-b) The AM uses table MAP_1 for s_i^j for each i and finds the values of $\tilde{s}_i^j$. Let $\widetilde{S}_j = (\tilde{s}_1^j, \ldots, \tilde{s}_m^j)$.

(Step 1-c) AM uses table MAP_3 for $d_j \otimes h_j$ and decrypts the output value. The reason MAP_3 is used here is to prevent AM finding any other information except $d_j \otimes h_j$ is more than $M + 1$ or not. If the output value is 1, the number of winners and survivors are more than or equal to $M + 1$. Then, AM updates

$W_j = W_{j+1} = (w_1^{j+1}, \ldots, w_m^{j+1})$
$C_j = \widetilde{S}_j = (\tilde{s}_1^j, \ldots, \tilde{s}_m^j)$
$z_{M+1st}^{(j)} = 1$

If the output value is 0, then

$W_j = W_{j+1} + S_j = (w_1^{j+1} \otimes s_1^j, \ldots, w_m^{j+1} \otimes s_m^j)$
$C_j = C_{j+1} - \widetilde{S}_j = (c_1^{j+1} \otimes (\tilde{s}_1^j)^{-1}, \ldots, c_m^{j+1} \otimes (\tilde{s}_m^j)^{-1})$
$z_{M+1st}^{(j)} = 0$

There is no case where $C_{j+1}[i] = 0$ and $\widetilde{S}_j[i] = 1$ for all players $(1 \leq i \leq m)$. Thus $C_{j+1}[i] - \widetilde{S}_j[i]$ can be properly calculated.

(Step 2) For the final $W_0 = (w_1^0, \ldots, w_m^0)$, AM decrypts each w_i^0 with verification protocols and obtains the winners of the auction. P_i is the winners if and only if plaintext of $w_i^0 = 1$ and $\sum_{i=1}^m w_i^0 = M$. The $M + 1st$ highest price is

obtained as $Z_{M+1st} = (z^{(k-1)}_{M+1st}, \ldots, z^{(0)}_{M+1st})_2$.

3.7 Security

1. **Privacy for bidding prices**

   Each player can not retrieve any information except for the winners and the $M+1st$ highest price. An auction scheme is secure if there is no polynomial time adversary that breaks privacy with non-negligible advantage $\epsilon(\tau)$. We prove that the privacy for bidding prices in the proposed auction protocols under the assumption that BGN encryption with the mix-and-match oracle is semantically secure. Given a message m, the mix-and-match oracle receives an encrypted value $x_1 \in E_{G_1}(m)$ and returns the encrypted value $x_2 \in E_G(m)$ according to the mix-and-match table shown in Table 7. (which has the same function as MAP_3).

   Given a message m and the ciphertext $x_1 \in E_{G_1}(m)$, the function of mix-and-match table is to map $x_1 \in E_{G_1}(m) \rightarrow x_2 \in E_G(m)$. The range of the input value is supposed to be $\{0,1,\ldots,m\}$ and the range of the output is $\{0,1\}$. We do not consider cases where the input values are out of the range. Using this mix-and-match oracle, an adversary can compute any logical function without the limit where BGN encryption scheme can use only one multiplication on encrypted values. MAP_1 can also be computed if the range of the input value is restricted in $\{0,1\}$. Here, we define two semantically secure games and advantages for BGN encryption scheme and the proposed auction protocols. We also show that if there is adversary $\mathcal{B}$ that breaks the proposed auction protocol, we can compose adversary $\mathcal{A}$ that breaks the semantic security of the BGN encryption with the mix-and-match oracle by using $\mathcal{B}$.

   Definition 3 Let $\Pi = (KeyGen, Encrypt, Decrypt)$ be a BGN encryption scheme, and let $A^{O_1} = (A_1^{O_1}, A_2^{O_1})$, be a probabilistic polynomial-time algorithm, that can use the mix-and-match oracle O_1.

$$
\begin{array}{c}
(PK, SK) \leftarrow KeyGen \\
(m_0, m_1, s) \leftarrow A_1^{o_1}(PK) \\
b \leftarrow \{0, 1\} \\
c \leftarrow Encrypt(PK, m_b) \\
b' \leftarrow A_2^{o_1}(c, s) \\
return\ 1 \text{ iff } b \ = \ b'
\end{array}
$$

Figure 8: $EXPT_{A,\Pi}$

$$
\begin{array}{c}
(PK, SK) \leftarrow KeyGen \\
(b_1, b_2, \ldots, b_{m-1}, b_{m_0}, b_{m_1}, s) \ \leftarrow \ B_1(PK) \\
b \leftarrow \ \{0, 1\} \\
c_1 \leftarrow Bid(PK, b_1), c_2 \leftarrow Bid(PK, b_2), \ldots, c_{m-1} \leftarrow Bid(PK, b_{m-1}), c_m \leftarrow Bid(PK, b_{m_b}) \\
(winner, winning\ price) \leftarrow \ WinnerDecision(c_1, c_2, \ldots, c_{m-1}, c_m) \\
b' \leftarrow \ B_2(winner, winning\ price, s, view_{WinnerDcision}) \\
return\ 1 \text{ iff } b \ = \ b'
\end{array}
$$

Figure 9: $EXPT_{B,\Pi}$

$$\text{BGN-}Adv(\tau) = \Pr[EXPT_{A,\Pi}(\tau) \ = \ 1] - 1/2$$

where, $EXPT_{A,\Pi}$ is a semantic security game of the BGN encryption scheme with the mix-and-match oracle shown in Fig. 8.

We then define an adversary $\mathcal{B}$ for an auction protocol and an advantage for $\mathcal{B}$.

Definition 4 Let $\Pi = (KeyGen, Bid, WinnerDecision)$ be a secure auction protocol, and let B be two probabilistic polynomial-time algorithm B_1 and B_2.

$$\text{Auction-}Adv(\tau) = \Pr[EXPT_{B,\Pi} \ = \ 1] - 1/2$$

where $EXPT_{B,\Pi}$ is a semantic security game of the privacy of the auction protocol shown in Fig. 9. Bid is the function of encrypting the bidding price of each player. $WinnerDecision$ is the function of executing the auction with encrypted bids in order to find the winner and winning price.

First of all, B_1 generates k-bit integers, $b_1, b_2, \ldots, b_{m-1}$ as plaintexts of bidding prices for player 1 to $m-1$, and two challenge k-bit integers as b_{m_0}, b_{m_1} where b_{m_0} and b_{m_1} are the same bits except for i-th bit m_0^i and m_1^i. We assume b_{m_0} and b_{m_1} are not the $M+1st$ highest price. Then the function Bid is used for encrypting players' bidding prices such as $(c_1 = Bid(PK, b_1), c_2 = Bid(PK, b_2), \ldots, c_{m-1} = Bid(PK, b_{m-1}), c_m = Bid(PK, b_{m_b}))$ where $b \xleftarrow{r} \{0,1\}$. Finally the auction is executed with the function $WinnerDecision(c_1, c_2, \ldots, c_{m-1}, c_m)$ as the players' encrypted bidding prices. After the auction, B_2 outputs $b' \in \{0,1\}$ as a guess for b. $\mathcal{B}$ wins if $b = b'$.

Theorem 3 The privacy of the auction protocols is secure under the assumption that the BGN encryption is semantically secure with a mix-and-match oracle.

Proof. We show if there is adversary $\mathcal{B}$ that breaks the security of the proposed auction protocol, and we can compose adversary $\mathcal{A}$ that breaks the semantic security of the BGN encryption with the mix-and-match oracle. $\mathcal{B}$ generates k-bit integers, $b_1, b_2, \ldots, b_{m-1}$ and two challenge k-bit integers as b_{m_0}, b_{m_1} where b_{m_0} and b_{m_1} are the same bits except for i-th bit m_0^i and m_1^i following the definition. $\mathcal{A}$ receives two challenge k-bit integers as b_{m_0} and b_{m_1} from $\mathcal{B}$ and then $\mathcal{A}$ uses m_0^i and m_1^i as challenge bits for the challenger of the BGN encryption. Then $\mathcal{A}$ receives c as a result of $Encrypt(PK, m_b^i)$ and send it to $\mathcal{B}$. $\mathcal{B}$ receives $c_1, \ldots, c_{m-1}$, and c as c_m as the result of function Bid and uses $WinnerDecision$ function to execute a secure auction protocol with the mix-and-match oracle.

When calculation of plain equality test or mix-and-match is needed such as checking whether h_j is 0 and updating W, $\mathcal{A}$ uses mix-and-match oracle to transfer encrypted value over E_{G_1} to E_G. b_{m_0} and b_{m_1} are not the winning bidding prices and $\mathcal{A}$ knows all the input values, $b_1, b_2, \ldots, b_{m-1}$ except the i-th bit of b_{m_b}. So, $\mathcal{A}$ with mix-and-match oracle can simulate an auction for the adversary of auction $\mathcal{B}$. Through the auction, $\mathcal{B}$ observes

the calculation of the encrypted values and the results of the auction. After the auction, $\mathcal{B}$ outputs b', which is the guess for b. $\mathcal{A}$ outputs b', which is the same guess with $\mathcal{B}$'s output for b_{m_b}. If $\mathcal{B}$ can break the privacy of the bidding prices in the proposed auction protocol with advantage $\epsilon(\tau)$, $\mathcal{A}$ can break the semantic security of the BGN encryption with the same advantage.

□

2. **Correctness**

   For correct players' inputs, the protocol outputs the correct winner and price. From Theorem 1 introduced in Section 3.2, the bit-slice auction protocol obviously satisfies the correctness.

3. **Verification of the evaluation**

   To verify whether the protocol works, players need to validate whether the AM decrypts the evaluations of the circuit on ciphertexts through the protocol. We use the verification protocols introduced above so that each player can verify whether the protocol is computed correctly. There is a case where PET fails with negligible probability as described in 3.1.3. However, the failure of PET brings the miscalculation of auction result. For example, if PET used for the transformation of s_j^i fails, it brings a false winner or loser. We assume that AM proceeds the auction properly with verification protocol, thus in that kind of case players can detect the failure of PET with verification protocol.

3.8 Comparison of auction protocols

3.8.1 First price auction

The protocol proposed [**?**] requires mk AND computations and k plaintext equality tests. One AND computation requires two plaintext equality tests. So, the total number of plaintext equality test is $mk + k$. On the other hand, We do not use mix-and-match protocols anymore. The proposed protocol is based on only a 2-DNF scheme. It requires k plaintext equality tests when it

	AND	PET	Total PET(approx.)
[**?**]	mk	k	$2mk + k$
Proposed	0	k	k

Table 8: The number of PET in the first price auction.

	AND	OR	$Select_m$	MAP_1	MAP_2	Total PET(approx.)
[**?**]	$(2m-1)k$	$(m-1)k$	k	0	0	$(13mk/2) - 4k$
Proposed	0	0	k	mk	k	$2mk$

Table 9: The number of PET in the second price auction.

checks whether b^i_{max} is the ciphertext of 0. A comparison between the proposed protocol and that in [**?**] is shown in the Table 8.

3.8.2 Second price auction

In the second price auction protocol, the protocol in [**?**] requires $(2m-1)$ AND, $(m-1)k$ OR and k $Select_m$ gates. One OR gate requires two plaintext equality tests. $Select_m$ gate has $2k+1$ input bits as below.

$Select_m(b, x^{(m-1)}, ..., x^{(0)}, y^{(m-1)}, ..., y^{(0)})$

It requires one test to check whether b is the ciphertext of 1, so in total approximately $(13mk) - 4k$ requires plaintext equality tests are required. Conversely, the proposed protocol requires MAP_1 mk times and MAP_2 k times. MAP_1 requires one plaintext equality test and MAP_2 requires approximately one half of m times on average, so in total $2mk$. A comparison between the proposed protocol and that in [**?**] is shown in the Table 9. In second price auction we can reduce the number of times when the plaintext equality test is executed.

3.8.3 $M+1st$ price auction

The protocol proposed in [**?**] is based on homomorphic encryption. In their protocol, bidding price of each player is not represented as binary bit. Therefore, for a potential bidding price p and m players, each player needs to execute encryption p times for bidding, and AM calculates multiplications of ciphertexts $2mp$ times to run the auction. PET (plaintext equality test) is used in the opening phase to check whether the number of i-th bid is more than $M+1$ or not with using binary search for each price i in $[1, p]$. Binary search for p needs

	[?]	Proposed
Bidding (per one bidder)	p encryptions	log p encryptions
Running auction (Calculation over group)	$2mp$ multiplications	$m$log p multiplications $m$log p pairing
Running auction (PET)	log $p(M+1)$ times	log $p(M+1) + mp$ times
Decrypting to decide the winners	m decryptions	m decryptions
Decrypting to decide the winning price	log p decryptions	log p decryptions

Table 10: The Comparison of computational complexity in $M+1$st price auction.

log p comparisons and one comparison needs PET $M+1$ times for each bid to check whether it is more than $M+1$. In the end of auction, m and log p decryptions are used to decide the winner and winning price of the auction.

Our auction protocol is based on BGN encryption where each player's bidding price is represented as a binary expression. We use PET mp times when AM calculates $\tilde{s}_i^j$ from player j's i-th bid for all i and j. We also use PET when AM detects whether $b_{M+1st}^{(i)}$ is more than M or not. log p decryptions are used to open the winning price and m decryptions are used to open the winners of auction. A comparison between the proposed protocol and that in [**?**] is shown in Table 10. Although the number of encryption and multiplication in the proposed protocol is reduced compared to the protocol in [**?**], the proposed protocol needs $m$log p paring calculation. The computation cost of paring calculation is approximately 4 times higher than that of group calculation in the worst case [**?**]. Therefore, for the evaluation of efficiency, the proposed protocol is certainly more efficient than that in [**?**]. As for the communication costs, communication during Bidding and Opening phase in [**?**] and proposed protocol is the same, so it depends on the encrypted message size(that is, proportional to the key size) of each protocol.

A secure auction protocol for the first and second price auction was introduced in subsection 3.4 and 3.5 respectively. However, in case of second price auction ($M=1$), the proposed protocol is approximately twice faster than the one in subsection 3.5. In order to obtain the second highest bidding price, the

protocol in [**?**] executes the first price auction protocol again after eliminating the highest bid.

Chapter 4 Punishment strategy and its problem

From cyber security perspectives, the Internet involves various players as to enterprises securing information and attackers aiming to steal such information. Various researches have been conducted to analyze such players' behavior in e-commerce transactions by applying game theory. This chapter discusses mechanisms which lead players to appropriately behave by a punishment strategy which poses a penalty to dishonest players. At first, application of cryptography to game theory, punishment strategy and its problems will be introduced, and then a definition to solve such problems will be proposed.

4.1 The Correlated element selection problem

In most common games, the joint strategy of the players is described by a short list of tuples $(move_1, move_2, ..., move_n)$, $(move'_1, move'_2, ..., move'_n)$, $(move''_1, move''_2, ..., move''_n)$ where the strategy is to choose at random one tuple from this list, and have Player 1 play $move_1$, Player 2 play $move_2$,..., Player n play $move_n$. (For example, in the two-player game of chicken the list consists of three pairs $(D, C), (C, D), (C, C)$.) Hence, to obtain an efficient solution for such games, we need an efficient cryptographic protocol for the following problem: Two players, A and B, know a list of pairs $(a_1, b_1), ..., (a_i, b_i), ..., (a_n, b_n)$ (maybe with repetitions), and they need to jointly choose a random index i, and have player A learn only the value a_i and player B learn only the value b_i. This problem called the Correlated element selection problem was introduced by Dodis, Halevi, and Rabin [**?**]. To describe an efficient solution for this problem, we first introduce some notations, tools, and an simple protocol that solves this problem in the special case where the players are "honest but curious", and then explain how to modify this protocol to handle the general case where the players can be malicious.

4.1.1 Notations and tools

We denote by $[n]$ the set 1,2,...,n. For a randomized algorithm A and an input x, we denote $A(x)$ is the output distribution of A on x, and $A(x; r)$ is

the output string when using the randomness r. If one of the inputs to A is considered a "key", then we write it as a subscript (e.g. $A_k(x)$). We use $pk, pk_1, pk_2, ...$ to denote public keys and $sk, sk_1, sk_2, ...$ to denote secret keys. The main tool that we use in our protocol is blindable encryption schemes. Like all public-key encryption schemes, blindable encryption schemes include algorithms for keygeneration, encryption and decryption. In addition they also have a "blinding" and "combining" algorithms. We denote these algorithms by $Gen, Enc, Dec, Blind$, and $Combine$, respectively. Below we formally define the blinding and combining functions. In this definition we assume that the message space M forms a group (which we denote as an additive group with identity 0).

Definition 5 (Blindable encryption). A public-key encryption scheme E is blindable if there exist (PPT) algorithms $Blind$ and $Combine$ such that for every message m and every ciphertext $c \in Enc_{pk}(m)$:

- For any message m' (also referred to as the "blinding factor"), $Blind_{pk}(c, m')$ produces a random encryption of $m+m'$. Namely, the distribution $Blind_{pk}(c, m')$ should be equal to the distribution $Enc_{pk}(m + m')$.
  $Enc_{pk}(m + m') \equiv Blind_{pk}(c, m')$
- If r_1, r_2 are the random coins used by two successive "blindings", then there exists an algorithm $Combine$ for any two blinding factors m_1, m_2,
  $Blind_{pk}(Blind_{pk}(c, m_1; r_1), m_2; r_2) = Blind_{pk}(c, m_1+m_2; Combine_{pk}(r_1, r_2))$

Thus, in a blindable encryption scheme anyone can "randomly translate" the encryption c of m into an encryption c' of $m + m'$, without knowledge of m or the secret key, and there is an efficient way of "combining" several blindings into one operation. The ElGamal encryption schemes can be extended into blindable encryption schemes. We note that most of the components of our solution are independent of the specific underlying blindable encryption scheme, but there are some aspects that still have to be tailored to each scheme.

4.1.2 A protocol for the honest-but-curious case

We introduce an efficient protocol in the case of honest-but-curious players. Let us recall the Correlated Element Selection problem. Two players share a public list of pairs $(a_i, b_i)_{i=1}^n$. For reasons that will soon become clear, we call the two

Common inputs: List of pairs $(a_i, b_i)_{i=1}^n$, public key pk

Prepare known: secret key sk

P: **1.Permute and Encrypt.**

Pick a random permutation π over $[n]$.

Let $(c_i, d_i) = (Enc_{pk}(a_{\pi(i)}), Enc_{pk}(b_{\pi(i)}))$, for all $i \in [n]$

Send the list $(c_i, d_i)_{i=1}^n$ to C

C: **2.Choose and Blind.**

Pick a random index $l \in [n]$, and a random blinding factor β

Let $(e, f) = (Blind_{pk}(c_l, 0), Blind_{pk}(d_l, \beta)))$.

Send (e, f) to P.

P: **3.Decrypt and Output**

Set $a = Dec_{sk}(e), \tilde{b} = Dec_{sk}(f)$. Output a.

Send $\tilde{b}$ to C.

C: **4.Unblind and Output**

Set $b = \tilde{b} - \beta$. Output b.

Figure 10: Protocols for Correlated Element Selection in the honest-but-curious model (CES-1),

players the "Preparer" (P) and the "Chooser" (C). The players wish to pick a random index i such that P only learns a_i and C only learns b_i. Figure 10 describes the Correlated Element Selection protocol for the honest-but-curious players. We employ a semantically secure blindable encryption scheme, and for simplicity we assume that the keys for this scheme were chosen by a trusted party ahead of time and given to P, and that the public key was also given to C. At the beginning of the protocol, the Preparer randomly permutes the public list of pairs $(a_i, b_i)_{i=1}^n$, encrypts them element-wise and sends the resulting list $(c_i, d_i)_{i=1}^n$ to the Chooser. (Since the encryption is semantically secure, the Chooser can not extract any useful information about the permutation π.) The Chooser picks a random pair of ciphertexts (c_l, d_l) from the permuted list (so the final output pair will be the decryption of these ciphertexts). It then blinds c_l with 0 (i.e. makes a random encryption of the same plaintext), blinds d_l with a random blinding factor β, and sends the resulting pair of ciphertexts (e, f) back to the Preparer. Decryption of e gives the Preparer its element a_l (and nothing more, since e is a random encryption of a_l after the blinding with 0), while the decryption $\tilde{b}_l$ of f does not convey the value of the actual encrypted message since it was blinded with a random blinding factor. The Preparer sends b to the Chooser, who recovers his element b by subtracting the blinding factor β.

It is easy to show that if both players follow the protocol then their output is indeed a random pair (a_i, b_i) from the known list. Moreover, at the end of the protocol the Preparer has no information about b other than what is implied by its own output a_l, and the Chooser gets "computationally no information" about a_l other than what is implied by b_l.

4.1.3 Dealing with dishonest players

Following the common practice in the design of secure protocols, one can modify the above protocol to deal with dishonest players by adding appropriate zero-knowledge proofs. That is, after each flow of the original protocol, the corresponding player proves in zero knowledge that it indeed followed its prescribed protocol: After Step 1, the Preparer proves that it knows the permutation π that was used to permute the list. After Step 2 the Chooser proves that it

knows the index l and the blinding factor that was used to produce the pair (e, f). Finally, after Step 3 the Preparer proves that the plaintext $\tilde{b}_l$ is indeed the decryption of the ciphertext f. Given these zero-knowledge proofs, one can appeal to general theorems about secure two-party protocols, and prove that the resulting protocol is secure in the general case of potentially malicious players. We note that the zero-knowledge proofs that are involved in this protocol can be made very efficient, so even this "generic" protocol is quite efficient. However, a closer look reveals that one does not need all the power of the generic transformation, and the protocol can be optimized in several ways. Some of the optimizations are detailed below. The resulting protocol CES-2 is described in Figure 11. Protocol CES-2 securely computes the (randomized) function of the Correlated Element Selection problem. To withstand malicious players, the Preparer P must "prove" that the element $\tilde{b}_l$ that it send in Step 3 of CES-1 is a proper decryption of the ciphertext f. However, this can be done in a straightforward manner without requiring zero-knowledge proofs. Indeed, the Preparer can reveal additional information (such as the randomness used in the encryption of f), as long as this extra information does not compromise the semantic security of the ciphertext e. The problem is that P may not be able to compute the randomness of the blinded value f (for example, in ElGamal encryption this would require computation of discrete log). Hence, we need to devise a different method to enable the proof. The proof will go as follows: for each $i \in [n]$, the Preparer sends the element $b_{\pi(i)}$ and corresponding random string that was used to obtain ciphertexts d_i in the first step. The Chooser can then check that the element d_l that it chose in Step 2 was encrypted correctly, and learn the corresponding plaintext. Clearly, in this protocol the Chooser gets more information than just the decryption of f (specifically, it gets the decryption of all the d_i's). However, this does not affect the security of the protocol, as the Chooser now sees a decryption of a permutation of a list that he knew at the beginning of the protocol. This permutation of the all b_i's does not give any information about the output of the Preparer, other than what is implied by its output b. In particular, notice that if b appears more than once in the list, then the Chooser does not know which of these occurrences was

encrypted by d_l. Next, we observe that after the above change there is no need for the Chooser to send f to the Preparer; it is sufficient if C sends only e in Step 2, since it can compute the decryption of d_l by itself.

4.1.4 Realizing correlated equilibrium with cheap talk

Consider some n-player game $\Gamma = (A_i,u_i)$ in normal form, along with a correlated equilibrium D. We then define the extensive form game Γ_{CT} in which all players first communicate in a cheap talk phase before the original game Γ. In [**?**], they showed correlated equilibrium can be realized by using the correlated element selection protocol in the extensive form game Γ_{CT}. Following the game-theoretic convention, all players must play some actions in Γ (i.e., we do not allow player P_i to abort in Γ unless this is an action in A_i). On the other hand, following the cryptographic convention we allow players to abort during the cheap talk phase.

Punishment strategy was suggested as a kind of rules that prevents players from aborting in the cheap talk phase. If a player aborts, the other players take actions that make aborting player's utility low. So all players do not have incentive to abort in the cheap talk phase and deviating from an action in the original game. The initial result of punishment strategy was shown in [**?**], that examines the case of two-player game. The basic idea is as follows: Let D be a correlated equilibrium in a two-player games Γ in Γ_{CT}, the two players run a protocol Π to calculate $(a_1, a_2) \leftarrow D$, where player P_i receives a_i as an output. This protocol Π is secure-with-abort (cf. [**?**]), which informally means that privacy and correctness holds, on the other hand, fairness does not; in particular, we assume it is possible for P_1 to receive its output even though P_2 does not. After running Π, each player plays the action it received as the output in Π; if P_2 does not receive an output from Π then it plays the minimax profile against P_1. The minimax profile against P_i is an action $a_{-i} \in A_{-i}$ that minimizes $max_{a_i \in A_i}\ u_i(a_i, a_{-i})$. Kats generalized this punishment strategy from two-player to n-player in [**?**]. Assume that some players select actions following the recommendation from the outputs of Π, while some collude with each other (which is called coalition C) and deviate from recommendation. C prefers σ to $\acute{\sigma}$ only if every player in C weakly prefers σ to $\acute{\sigma}$ and some player in C strictly

Common inputs: List of pairs $(a_i, b_i)_{i=1}^n$, public key pk

Prepare known: secret key sk

P: **1.Permute and Encrypt.**

Pick a random permutation π over $[n]$.

Let $(c_i, d_i) = (Enc_{pk}(a_{\pi(i)}), Enc_{pk}(b_{\pi(i)}))$, for all $i \in [n]$

Send the list $(c_i, d_i)_{i=1}^n$ to C

Sub-protocol Π_1: P proves in zero knowledge that

it knows the randomness $(r_i, s_i)_{i=1}^n$

and permutation π that were used to obtain the list $(c_i, d_i)_{i=1}^n$.

C: **2.Choose and Blind.**

Pick a random index $l \in [n]$, and a random blinding factor β

Let $(e, f) = (Blind_{pk}(c_l, 0), Blind_{pk}(d_l, \beta)))$.

Send (e, f) to P.

Sub-protocol Π_2: C proves in a witness-independent manner

that it knows the randomness and index l that were used to obtain e.

P: **3.Decrypt and Output**

Set $a = Dec_{sk}(e), \tilde{b} = Dec_{sk}(f)$. Output a.

Send $\tilde{b}$ to C.

C: **4.Unblind and Output**

Set $b = \tilde{b} - \beta$. Output b.

Figure 11: Protocols for Correlated Element Selection (CES-2),

prefers σ to $\acute{\sigma}$.

Definition 6 Let Γ be an n-player game with correlated equilibrium D. A strategy vector ρ is a t-punishment strategy with respect to D if for all $C \subseteq$ N with $\mid C \mid \leq$ t, and all $\acute{\sigma}_C$ it holds that for all $i \in C$, $u_i(\acute{\sigma}_C, \rho_{-C}) \leq u_i(D)$.

We introduce another definition of punishment strategy in [**?**]. In [**?**] they considered the cases when there are Byzantine failure players. They defined a protocol is k-immune if the protocol tolerates to at most k Byzantine failure players. If any set of players T whose size is at most k cannot give the rest of players a worse payoff, even if the players in T can communicate with each other. For simplicity of discussion, this thesis assumes that k=0, that is, there is no Byzantine failure players. They also consider type t_i which is an input given to each player at the beginning. This thesis does not consider type t_i, that is, there is a single type for every player. The example in this thesis can be easily extended to the cases where there are multiple types for players.

Definition 7 If Γ is an underlying game with a mediator M, a strategy profile ρ in Γ is a t-punishment if for all subsets $C \subseteq N$ with $\mid C \mid \leq$ t, all strategies σ in Γ with a cheap talk CT(C) among players *in* C, and all players $i \in C$, $u_i(\Gamma, \sigma) > u_i(\Gamma + CT(C), \sigma_C, \rho_{-C})$.

A remarkable difference between Definition 6 and Definition 7 is allowing equal utilities or not. In this meaning, Definition 7. requires the stronger condition. Intuitively, for any set C, even if all players in C collude and communicate each other with the cheap talk, no player in C can obtain a better payoff than the correlated equilibrium if the rest of the players select the punishment strategy. In [**?**], they showed that for six-player games with a 2-punishment strategy, any Nash equilibrium can be implemented even in the presence of at most one malicious player.

4.2 Cheating players' actions against punishment strategy

This section shows an example that the punishment strategy does not prevent the players in C aborting in the cheap talk phase. The example is shown in Figure 12. We consider a 5-player game with 2 malicious players. That satisfies

the conditions in both in [?] and [?] introduced above. However, a table to show a 5 players game is so complicated to explain, so to simplify the example, we introduce a dummy player defined as below.

Definition 8 A dummy player (P_d) is the player who satisfies these conditions,

1. His action does not effect to the other players' utilities.
   $\forall\sigma_d,\ \acute{\sigma_d}\ \in A_d,\ \forall\sigma_{-d} \in \prod_{-d}$ (the set of the parties other than the dummy player).
   $$u_{-d}(\sigma_d, \sigma_{-d}) = u_{-d}(\acute{\sigma_d}, \sigma_{-d}).$$
2. His utility is not effected from the other players' actions except for the case when a punishment strategy is taken,
   $\forall\sigma_d,\ \in A_d,\ \forall\sigma_{-d} \in \prod_{-d}$ and a punishment strategy ρ_{-d},
   $$u_d(\sigma_d, \sigma_{-d}) < u_d(\sigma_d, \rho_{-d})$$
   otherwise, $\forall\sigma_{-d},\ \acute{\sigma_{-d}}\ \in \prod_{-d} - \{\rho\},\ \forall\sigma_d \in A_d$
   $$u_d(\sigma_d, \sigma_{-d}) = u_d(\sigma_d, \acute{\sigma_{-d}})$$

$N = \{P_1, P_2, P_3, P_4, P_5\}$, we assume P_5 is a dummy player, so we do not care about his action here. The number of malicious players is 2 (t=2), and for $1 \leq i \leq 4$, P_i's action set is $A_i = \{a_1^i, a_2^i, a_3^i, a_4^i\}$. The utility u_i is shown in Figure 12. Figure 12 consists of 4×4 sub-tables. The utilities when P_4 takes a_i^4 and P_3 takes a_j^3 is shown in the sub-table at i-th row and j-th column. In each subtable, P_1's action is listed in the low and P_2's action is listed in the column. Each entry is the tuple of utilities, (u_1, u_2, u_3, u_4). The correlated equilibria for this game are $(a_3^1, a_3^2, a_3^3, a_2^4)$ and $(a_1^1, a_1^2, a_2^3, a_3^4)$. In these cases, the utilities of the players are (5,5,5,5), shown by the bold boxes.

Let us consider the case when P_3 and P_4 aborts in the cheap talk phase. After aborting the protocol, they declare that they will take actions a_1^3 and a_1^4 using the cheap talk, the rest of players are supposed to select the punishment strategy (a_4^1,a_4^2) and as a result, the set of actions is (a_4^1,a_4^2,a_1^3,a_1^4), then each player will receive a payoff (u_1,u_2,u_3,u_4) = (3,3,3,3). P_3' and P_4' payoffs decrease from the correlated equilibria. So, these payoffs satisfy the definition of a punishment strategy (for all $C \subseteq$ N and all $\acute{\sigma}_C$ it holds that for all $i \in C$ $u_i(\acute{\sigma}_C, \rho_{-C}) \leq$

$u_i(D)$).

In game theory, all players are considered to be rational, so if there is a better set of actions for P_1 and P_2, it is natural for them to select a better action than Nash equilibrium. The rest of players P_1 and P_2 know actions which P_3 and P_4 take and their utilities when they select punishment strategy. They know P_3 and P_4 are rational and on the other hand, P_1 and P_2 know they are rational. The utility for (P_1, P_2) of the punishment strategy is worse than that of the other strategies. If the players are honest, they will select the punishment strategy even if they receive worse payoff than the other strategies. However, the players are rational and all players know they are rational. Thus the aborting players think they will not execute the punishment strategy. This is called as "empty threat" [**?**].

In this example, In P_1's veiw, a_2^1 is the dominant strategy given that P_3 and P_4 take a_1^3, a_1^4, so P_3 and P_4 will think P_3 take action a_3^3. And given a set of actions (a_2^1,a_1^3,a_1^4), a_2^2 is the dominant strategy for P_2. So all players try to receive the maximum profit under the assumption that all players are rational, $(a_2^1,a_2^2,a_1^3,a_1^4)$ is the equilibrium for all the players. And in this case, even if P_3 and P_4 abort in the cheap talk phase, the other players will not punish them, rather help them for receiving more payoff than the punishment strategy. This is shown as an arrow in Figure 12. The players will select the set of actions $(a_2^1,a_2^2,a_1^3,a_1^4)$, not the punishment strategy $(a_4^1,a_4^2,a_1^3,a_1^4)$. In short, when P_3 and P_4 abort and declare that they will take better actions for them than correlated equilibrium, P_1 and P_2 are supposed to take the punishment strategy against P_3 and P_4 even the payoffs of P_1 and P_2 reduce. However, P_1 and P_2 are rational, they select actions that give them more payoffs than punishment strategy.

4.3 New definition of punishment strategy

The reason the punishment strategy does not work is that the definitions in [**?**] and [**?**] do not care about punishing players' utilities. So, it is natural for punishing players to take actions that give them better payoff. Otherwise punishment strategy could be "empty threat", as in [**?**] they showed a similar case for two-player games, a min-max strategy may be "empty threat" without proper

setting. For multiple player games, the above example shows that a punishment strategy does not work. To avoid the cases shown above, we suggest new definition of a punishment strategy which considers punishing players' utilities.

Definition 9 Let Γ be an n-player game with correlated equilibrium D. A strategy vector ρ is a t-punishment strategy if for any strategy vector $\acute{\rho}$ with respect to D and for all $i \in C \subseteq \mathrm{N}$, $j \notin C$ with $\mid C \mid \leq \mathrm{t}$, all $\acute{\sigma}_C$ it holds that
$u_i(\acute{\sigma}_C, \rho_{-C}) \leq u_i(D)$ and
$u_j(\acute{\sigma_C}, \acute{\rho}_{-C}) \leq u_j(\acute{\sigma}_C, \rho_{-C})$,
where $\acute{\sigma}$ satisfies the condition $u_i(D) \leq u_i(\acute{\sigma}_C, \rho''_{-C})$ for any strategy vector ρ''_{-C}.
We add new setting about the punishing players' utilities u_j to the original definition of punishment strategy.
The condition, $u_j(\acute{\sigma_C}, \acute{\rho}_{-C}) \leq u_j(\acute{\sigma}_C, \rho_{-C})$ is the setting for the punishing players to weakly prefer punishment strategy than the other strategies. By setting this, we can avoid the case where punishing players' utilities decrease when they punish aborting players.

Theorem 4 Let Γ be an n-player game with correlated equilibrium D and a punishment strategy as defined above. Correlated equilibria can be implemented even in the presence of at most t malicious players under the condition $n > 2t$.

Proof When some players C abort in the cheap talk phase, the other players are trying to punish them with a punishment strategy. Since all punishing players' utilities for the punishment strategy are not worse than the other strategies, they will select the punishment strategy. Malicious players know that the rest of players will choose the punishment strategy whenever they abort the protocol, they are not supposed to deviate from the protocol.

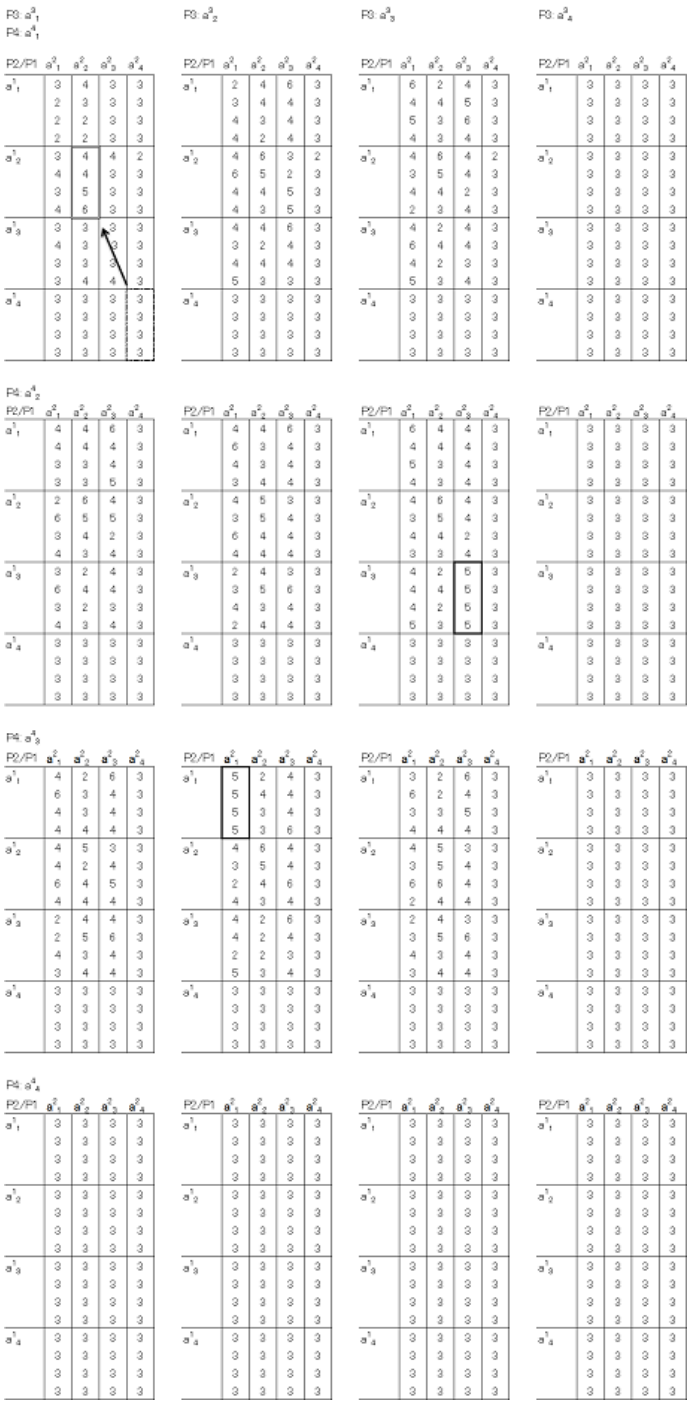

Figure 12: An example the case when the punishment strategy does not work.

Chapter 5 Proposed authentication method for web application

This chapter is to propose a new simple authentication method which has the same security level as two-factor authentication. With the proposed system, additional settings, installation or devices are not required to maintain accessibility for users. This method provides a digital signature and verification mechanism based on public key cryptosystem. On user side this system requires only web browser with JavaScript and HTML5 functions. With modern browsers such as Internet Explorer, Google Chrome and Firefox, JavaScript and HTML5 function are available by default. Consequently, user accessibility is kept since users are just required to input passwords as before. The characteristics of this system compared to the precedents are as follows:

1. This proposal is not for a password management tool but a system for login functions, and it is applicable by giving modifications to existing web application. The modifications do not require developing large-scale program but additional one column in a database for credentials and simple implementation for authentication method on server side.
2. The proposed method does not depend on OS on users' devices as long as the browser is compatible with HTML 5.
3. It reduces the risks of password leakage by automatically issue credentials for each web service.

5.1 Related Methods

5.1.1 Anomaly Detection

As a method to detect unauthorized login attempts for web application, anomaly detection has been used since before. Many anomaly detection techniques have been developed to to find unexpected patterns [**?**, **?**]. However, this is not adequate in terms of countermeasures against list-based attacks which is becoming more sophisticated.

1. Login attempts to a single account

   This is to temporary suspend accounts if login errors for the same account

are detected more than a certain number of times. This method can detect continuous attack to a certain accounts. However, since list-based attacks are conducted by referring to lists of credentials, such repeated attempts are not observed often. Therefore, it is difficult to detect unauthorized login attempts using this method.

2. Repeated login attempts from suspicious IP addresses

   This method temporarily prohibits login from a certain IP address if there are multiple login errors from the same IP address. With this method, it is possible to detect an attack when a certain IP address causes login errors. However, even if this measure is taken, there are attackers who change their IP address to continue such attacks. It is also possible to detect false positive in case where multiple login errors coincidentally occur from an IP address used as a gateway to the Internet at a large-sized company for instance.

5.1.2 Password Management Tool

Password management tool is designed to centrally handle passwords in order to reduce burden of remembering multiple passwords, including "1password"[?] and other products. While it only asks users to remember a master password, security of all the passwords is not guaranteed once the master password is broken. Also, if the user forgets the master password, all the credentials managed by the tool will be lost. In order to address this issue, Morii and others proposed and implemented a system to retrieve credentials using multiple secret questions [?]. By applying secret sharing in master password management, this proposal has realized a password management system that is more secure.

5.1.3 Two-Factor Authentication

Two-factor authentication refers to a system which verifies if the identification was provided from the user's device in addition to password authentication. As demonstrated in Figure 13, after an authentication with ID and password succeeds, users will be requested to input the authentication code sent to a registered device (e.g. SMS to a mobile phone). This prevents an unauthorized use of accounts even if the credentials are stolen. The authentication code changes with time and for each login attempt, and it is difficult for attackers

to predict. In the first stage, it verifies the authentication is provided by the user themselves by the ID and the password, and the second stage verifies that the authentication is provided from a registered device. The Two-factor authentication system is applicable for some web services, however, since its configuration and usage flow are complicated, it does not seem to be commonly employed. Fujikawa and others proposed and implemented a system with a smart phone application which has a function to automatically issue/manage passwords [**?**]. This enabled automatic login to websites using a smart phone as a key. However, methods to retrieve passwords if the phone is lost remained as an issue. With the increasing share of mobile devices, there are researches using mobile devices as the factor of authentication [**?**, **?**, **?**].

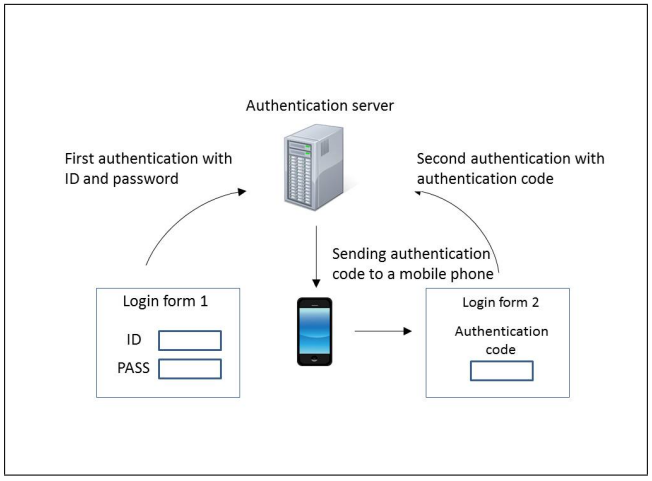

Figure 13: Flow of Two Factor Authentication.

5.1.4 Client Certification

Client certification can easily identify users since it uses public key certification installed on web browsers and does not ask for a credential each time. However, as a certification needs to be issued, this requires costs on both sides: service providers have to bear costs for distribution of client certifications especially for popular web services, and each user has to implement certification on their devices. For these reasons, client certification is not yet commonly applied as an authentication system for web services. Kobayashi and others proposed an authentication system based on client certification and FakeBasic [**?**]. Since

this system identifies users not by passwords but by a client certificate, users do not need to remember any password. However, costs for generating a client certificate on the servers and its implementing on users' side are the remaining issues.

5.1.5 SSH Public Key Authentication

Proposed method aims to realize secure web authentication by expanding SSH public key authentication for web services. We introduce SSH public key authentication as a basic idea of proposed method.

Authentication

1. A client sends a request to an authentication server.
2. The authentication server creates a random number and encrypts it with the public key.
3. The authentication server sends the encrypted random number to the client.
4. The client decrypts the random number with the secret key to retrieve the original random number. Then it applies a hash function to the random number.
5. The client sends the random number with the hash function applied to the authentication server.
6. The authentication server verifies the value sent from the client and the value gained by Step 2 with the hash function applied.
7. If the verification succeeds, the authentication is successful.

We propose a securely-enhanced method by applying this SSH public key authentication system in web applications.

5.1.6 SALT

Passwords saved in a server may be abused for unauthorized login if the server is intruded and the stored credentials are exposed. To prevent this, passwords are sometimes protected by hashing that are saved in a database. Since short or easy passwords are vulnerable to dictionary attacks and brute-force attacks, giving a SALT (some additional letters provided for each user's password) then hashing is a recommended procedure. Giving a SALT makes passwords longer and harder for attackers to retrieve the original password since a different hash value is given even if the same password is selected among several users. With

the proposed method, unauthorized login with leaked credentials from servers cannot be conducted. However, considering that it will take a long time until the proposed method becomes widely available, in this implementation, the passwords stored in a database are handled based on traditionally used SALT and hashes. According to RFC2898 [**?**], SALT is not confidential information, it can be uniquely generated from its ID. The proposed method applies an algorithm to uniquely generate SALT from the ID on servers and clients.

5.2 Registration and Authentication

The protocols for the registration and authentication between the client and the server is as follows. $Sign()$ and $Verify()$ refer to the signature and verification based on a public key encryption system, and $H()$ refers to a hash function. The overview of the registration flow and authentication flow are indicated in Fig. 14 and 15.

1. A client inputs an ID and password (PW). Simultaneously, it creates a pair of public key (PK) and secret key (SK) and saves SK in a client device. Even for the same client, a different pair of PK and SK is generated for each service.
2. The client sends ID, PW and PK to authentication server by using a secure route (e.g. SSL).
3. An authentication server saves ID and PK. It also generates $SALT$ from the ID, and saves the value ($H(PW||SALT)$), derived by hashing to the result of combining $SALT$ and PW.

Authentication

1. A client sends a request for authentication to the authentication server.
2. The authentication server generates a random number r.
3. The authentication server sends r to the client.
4. The client inputs ID and PW. It combines $SALT$ generated from ID and performs a hash function ($H(PW||SALT)$). Combining the results and r, and then a hash function is performed. Then a signature is provided for $H(H(PW||SALT)||r)$ using SK to generate σ. ($\sigma = Sign(SK, H(H(PW||SALT)||r))$)
5. The client sends ID and σ to the authentication server.

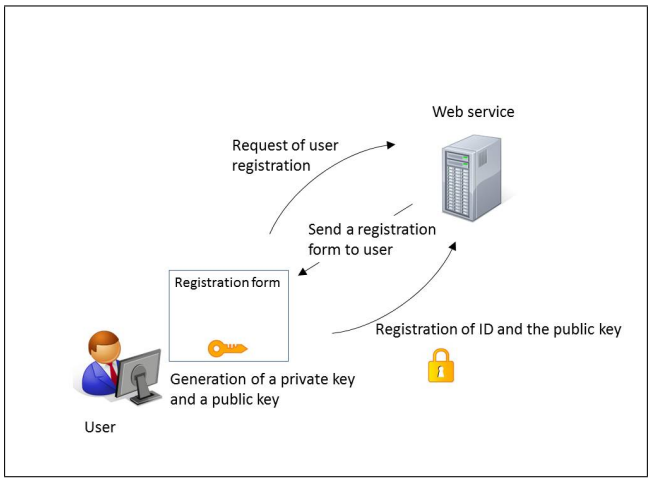

Figure 14: Registration.

6. The authentication server verifies σ using PK. $(Verify(PK, H(H(PW||SALT)||r), \sigma) = 1/0)$
7. If the signature verification succeeds, the authentication is successfully done.

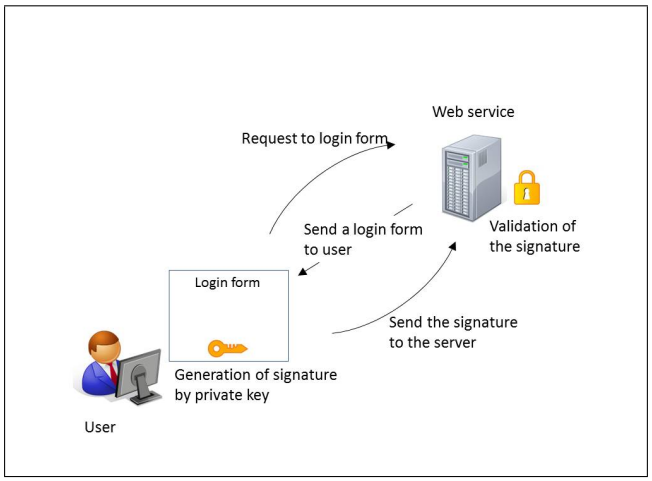

Figure 15: Authentication.

5.3 Implementation

The proposed technique is implemented in a web browser and a web server. As well as usual password authentications, an interface for users to input their ID and password on a browser is created. Generation of public keys and secret

keys, signature and its verification process for the purpose of this implementation are all automatically processed by JavaScript. Similarly, loading and reading secret keys saved in a browser's Web Storage area is also automatically processed by JavaScript. Therefore, users can realize the proposed method just by inputting credentials without any special procedure. In implementation of proposed method, we use ECDSA for signing and SHA-256 for hash functions.

5.3.1 Technical Components

This section discusses Web Storage and encryption library used for the implementation.

5.3.2 Web Storage

Web Storage [**?**] is a function introduced on HTML 5 which saves data accessible from JavaScript on the client. The limit of saving data is recommended for browsers to be at maximum 5MB per origin. There are two types of data: sessionStorage which are deleted at the end of each session and localStorage which remains even after the browser is shut down. For both cases, reading and writing function has to be done from the same origin, and therefore JavaScript from the other origins cannot access the data. In this thesis, localStorage is used since the data stored in Web Storage needs to be kept even after the browser is closed.

5.3.3 JavaScript Libraries

In order to process encryption on a user's browser, JavaScript library is used. For ECDSA key and signature generation on the client side, jsrsasign [**?**] is applied. Also, CryptoJS [**?**] is used for SHA-256, which is a hash function.

5.3.4 Implementation Format

Registration function and authentication function are implemented in order to realize authentication between a web server and a browser based on the proposed method. The screen that a user see when registering and authenticating is the one asking for credentials, same as in password authentication method. A registration screen and authentication screen is indicated in Fig. 16 and 17. Also, Fig. 18 shows an example of a secret key saved in Web Storage.

Registration

1. A User U_i inputs ID_i and a password (PW_i) and generates a pair of public

Figure 16: Registration form.

Figure 17: Login form.

and secret key PK_i, SK_i. SK_i is saved in the browser's Web Storage.

2. U_i sends ID_i, PW_i and PK_i to a server over secure channels such as SSL.
3. A web server saves ID_i and PK_i on its database. In order not to save PW_i in plain text format, $SALT_i$ is created from ID_i, which is combined to the original PW_i and a hash function is performed. $(H(PW_i||SALT_i))$ and ID_i are saved in the database.

Authentication

1. U_i creates a request for a login page to a web server.
2. The server creates a random number r and sends it together with a login page.
3. U_i inputs ID_i and PW_i on the login page. $SALT_i$ is created from ID_i, and SK_i is loaded from the browser's Web Storage. A signature is given to $H(H(PW_i||SALT_i)||r)$ to make σ $(\sigma = Sign(SK_i, H(H(PW_i||SALT_i)||r)))$
4. U_i sends ID_i, r and σ to the server.
5. On the web server, using by $H(PW_i||SALT_i)$ and r corresponding to ID_i,

σ is verified. $(Verify(PK_i, H(H(PW_i||SALT_i)||r), \sigma) = 0/1)$.

6. If the signature verification succeeds, authentication is successfully done.

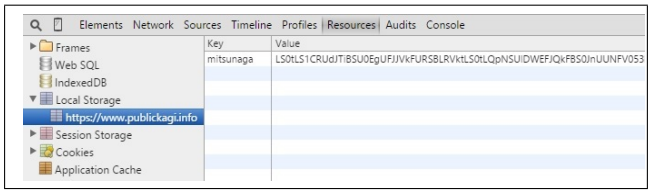

Figure 18: Secret Key inside Web Storage.

5.4 Signature Format

For the purpose to confirm the feasibility of proposed method, implementation was done using ECDSA signature method with elliptic curves. ECDSA consists of signature creation algorithm executed by a signer (a browser), signature verification algorithm executed by a verifier (a server), a setup to realise these algorithm and procedures to generate keys. The following indicates the details of each algorithm. A signer and a verifier are assumed to have shared elliptic curve parameters beforehand $(p, G = (x, y), n, h)$.

5.4.1 Key Generation

A signer generates a secret key and a public key by the following procedures:

1. A signer randomly chooses a secret key $d \in [1,n-1]$ using elliptic curve parameters and generates a public key $Q = dG$.
2. A verifier obtains the public key Q generated by the signer.

5.4.2 Signature Generation

A signer executes the following algorithm to generate a signature by using keys created by the setup and the key generation procedures (d, Q), elliptic curve parameters and a hash function H.

Input: plaintext M

Output: Signature for plaintext M $\sigma = (r, s)$

Algorithm:

1. Choose $k \in [1,n-1]$ randomly.
2. Calculate $kG = (x_1, y_1)$ and convert x_1 to an integer x_1'.

3. Calculate $r = x_1$' mod n. Return to step 1 if $r = 0$.
4. Calculate $H(M)$ and convert the output into bit stream $m \in$ [1,$n-1$].
5. Calculate k^{-1} mod n.
6. Calculate $s = k^{-1}(m + dr)$mod n. Return to step 1 if $s = 0$.
7. Output signature $\sigma = (r, s)$.

5.4.3 Signature Verification

A verifier executes the following algorithm to verify the signature using the public key generated by the setup and the key generation procedures, elliptic curve domain parameters and a hash function $H()$.

Input: plaintext M and the signature $\sigma = (r, s)$ for M.

Output: Signature verification successful or failed Algorithm:

1. Verify that r and s are integers as in [1,$n-1$].
2. Calculate $H(M)$ and convert the output into bit stream $m \in$ [1,$n-1$].
3. Calculate $u_1 = ms^{-1}$ mod n, $u_2 = rs^{-1}$ mod n.
4. Calculate $R = (x_r, y_r) = u_1G - u_2Q$. If $R = O$, output " signature verification failed ".
5. Convert x_r into an integer x_r', and calculate $v = x_r$' mod n.
6. If $v = r$, output " signature verification successful ". Output " signature verification successful " if $v \neq r$.

5.4.4 Observation

This system enables secure authentication for web services without reducing users' accessibility. Since a different secret key is generated and saved for each web service, accounts are not affected by unauthorized login even if the same password is shared for multiple services. Therefore, this method allows easier and more secure login for users compared to the traditional password login method without having to remember separate credentials for each web service. One of the issues for this system is that users need to change the pair of public and secret key when replacing devices since the secret key is stored within the browser in the device. This problem can be solved by using two-factor authentication by email; since many web services use an email address as an ID, for instance an additional authentication code can be issued to the email when registering a new key and identify the user's device in order to maintain

the key. Furthermore, if the old keys are not to be expired, authentication can be available from multiple devices, which allows users to use multiple devices at the same time.

5.5 Security

This section discusses the security issues using the proposed system related to information leakage from authentication server including passwords.

1. **Information Leakage from Authentication Server**

   An authentication server stores each user's ID, its corresponding public key (PK), and the hash value of the password using $SALT(H(PW||SALT))$. A situation can be assumed where an attacker who took control over an authentication server (e.g. a server's unauthorized admin) attempts to conduct unauthorized log in impersonating a legitimate user. Since the secret key SK is stored in the user's device, security is assumed to be guaranteed in this circumstance. ECDSA, applied in the proposed method, is proved to have Existential Unforgeability against Chosen Message Attack (EU-CMA) under generic oracle models [**?**]. Therefore, SK cannot be leaked from the server even if the attacker obtains the data in the server $(ID, PK, H(PW||SALT))$ and communication log related to authentication (r and σ). Consequently, the attacker is not able to create signature σ' impersonating a legitimate user.

2. **Password Eavesdropping**

   Security against eavesdropping on the communication is also considered. Essentially, passwords are communicated on a secure route such as SSL. This section discusses the possibility of password leakage in insecure environment without SSL etc. Communication logs related to authentication (r, ID, and σ) is assumed to be eavesdropped by an attacker. r is randomly created for each authentication instance. Also, σ is also a random number since it is generated based on $H(H(PW||SALT)||r)$. Therefore, it is difficult for an attacker to steal a password to log in to an account

impersonating a legitimate user, even if the passwords are eavesdropped.

3. **Leakage of Secret Key via Authentication Server**
   In the proposed method, secret keys are handled on JavaScript. If an attacker intrudes into an authentication server or modify JavaScript source code by XSS attacks etc., a user's secret key may be leaked when accessing to an authentication server. This issue can be prevented if a secret key are encrypted by symmetric key encryption with user's inputs as a key. However, this measure requires further discussion since it may impact users' accessibility.

5.6 Summary

In this chapter, an authentication technique based on public key using Web Storage, which is resistant to list-based attack was introduced. This chapter also discussed the security of the proposed method compared to password authentication, and it is proved that this method is more secure than the traditional method. It is suggested that this method can be eventually used as a secure authentication platform. Further information on the implementation is available on the following websites: (as of 30 October, 2015)

Source code https://github.com/sisoc-tokyo/pubkey-auth-demo

Demonstration https://secure.publickagi.info/demo

Chapter 6 Conclusion

This thesis proposed protocols for e-commerce transactions based on cryptography with the aim to realize such transactions which are safe from multiple perspectives. Instead of focusing on a single circumstance, this research divides issues to be considered for constructing e-commerce transactions into 3 layers and aims to construct a secure model for each layer.

As a background of this research, the situation surrounding e-commerce transactions was introduced in Chapter 1. Backed up with the prevalence of the Internet, opportunities for e-commerce transactions including Internet banking and online auctions have been developed, which has contributed to enhanced user convenience. Despite the advantage, however, risks of cyber threat have been increasing. Cyber attacks such as network intrusion, information theft and defacement have been continuously reported, and security measures for such issues are in demand. E-commerce is not an exception in this regard; incidents such as unauthorized online banking transfers and credit card usage on online shopping sites have been causing a large amount of damage in these years. Secure e-commerce transaction protocols should be implemented so that users can enjoy transactions without security concerns. In order to examine existing issues and propose solutions to overcome them, this research took a multi-layer approach by breaking down the challenges into three areas:

1. Consensus building on e-commerce transaction methods
2. Designing protocols for secure e-commerce transactions
3. Implementing application secure e-commerce transactions

By dividing the issues into the 3 layers and discuss solutions in each layer, this research aimed to propose a model for auctions that is secure from multiple perspectives. Since auctions involve more complex protocols compared to the other types of e-commerce transactions, the proposed model is expected to be applied in wider areas of e-commerce transactions as well.

Chapter 2, as preliminaries, introduced some basic terms of cryptography and game theory that appear in this thesis. This includes public key encryption and signature, Public Key, hohomorphic encryption, Infrastructure (PKI), key

sharing and game theory.

In Chapter 3, to design secure auction protocols, as the second approach, this thesis also proposed applying homomorphic encryption to "bit-slice" auction method. This proposal aims to solve privacy and fairness issues, which are inherent to auctions. Since the suggested encryption method allows processing encrypted numbers, this can be applied to auction protocols to process bidding prices while encrypted. Consequently, since the original bidding prices remain confidential, the auction will be carried out without the risk of data manipulation or privacy invasion. Further expanding this proposal, the chapter continued to prove that the proposed protocol is applicable to different types of auctions as well: First price auction, Second price auction and M+1 price auction. It also demonstrated that the proposed protocol is more efficient than the existing protocols in terms of processing costs.

As a first approach to set up a secure auction protocol, Chapter 4 considered players' behavior based on game theory and proposed a new punishment strategy. The chapter examined the issues with the existing punishment strategies and occasions where such strategies would not function. In some cases, it is possible that the incentives for players to jointly pose punishment to a dishonest player can be lower than the potential benefit that they may be able to gain by assisting a dishonest act. By further developing game theory, a new punishment strategy was proposed which includes settings to prioritize honest players' utilities. This improvement intends to further strengthen the role of the punishment strategies and to reduce the probability of players' dishonest behavior for securing auction protocols.

Finally in Chapter 5, a new authentication method for web applications was presented as a measure to securely implement the proposed protocols. This is based on the fact that e-commerce transactions are mostly operated through web applications which require user authentication. However, there are challenges in terms of security such as unauthorized login to user accounts. One of the attack methods is "list-based" attack, where leaked credentials are leveraged by threat actors. Although various mitigation efforts are in place, still there are some remaining issues with security and user convenience; securer measures

tend to affect user convenience for implementation. This chapter introduced a simple but also secure authentication method for web applications by using public key encryption. The method can be applied easily both for developers and users. Only some slight modifications are required for a credential database in a web server on the developer's side, and there is no additional procedure for users. Since a separate secret key is provided for each web service, it is thought that this method has resistance against list-based attacks even if a user sets the same credentials for different web services. This also saves users effort to create and remember different sets of credentials for every web services they use.

To this end, this thesis provides observations on potential security issues in comparison to the proposed authentication method and proves that the method is secure enough to remove such risks. Through the series of research, the thesis aimed to propose secure protocols for e-commerce. The proposal includes improving the existing punishment strategy for auctions based on game theory in order to efficiently prevent players' dishonest behavior, which is expected to contribute to securing auction protocols. As a secure system design, the thesis also proposed implementing encryption to auction protocols. Through this process, privacy and fairness were proved to be maintained by keeping the bidding prices encrypted. Finally, for secure implementation of e-commerce transactions on the Internet, a new authentication method was proposed. Supported by public key encryption, this proposed method is effective in protecting web service user credentials from being leaked or even leveraged. Despite the higher level of security compared to existing security measures around password, the proposed method still maintains user convenience and can be implemented effortlessly. As a final note, by combining the proposed protocols, it is expected that auctions can be conducted in a more securely compared to current protocols. Furthermore, since auctions usually involves more complex protocols compared to other e-commerce transactions, the proposed protocols can be applied to a wider types of e-commerce transactions. With the proposed enhanced security features, it is also predicted that it will contribute to reduce the risks of cyber attacks related to e-commerce transactions. In such a more secure environment, user trust in e-commerce transactions will be enhanced, which will

possibly contribute to the further expansion of the market and also to economic benefit.

List of Publications

Journal Paper

- Takuho Mitsunaga, Yoshifumi Manabe, and Tatsuaki Okamoto, "Efficient Secure Auction Protocols Based on the Boneh-Goh-Nissim Encryption", IEICE Transactions on Fundamentals, Vol.E96-A, No.1, pp.68-75, 2013.

International Conferences

- Takuho Mitsunaga, Yoshifumi Manabe, and Tatsuaki Okamoto, "A Secure M+1st Price Auction Protocol based on Bit Slice Circuits", Proceedings of IWSEC 2011, LNCS Vol. 7038, pp.51-64, 2011.
- Takuho Mitsunaga, Yoshifumi Manabe, Tatsuaki Okamoto, "Efficient Secure Auction Protocols Based on the Boneh-Goh-Nissim Encryption", Proceedings of IWSEC 2010, LNCS Vol. 6434, pp.149-163, 2010.
- Takuho Mitsunaga, Yoshifumi Manabe, Tatsuaki Okamoto, "Insatabity of A Punishment Strategy in Correlated Equilibria", Proceedings of Workshop on Algorithmic Game Theory: Dynamics and Convergence in Distributed Systems, AlgoGT, 2010.

Domestic Conferences

- Takuho Mitsunaga, Hiroshi Kobayashi, Yoshinori Matsumoto, Tomoyuki Shigemori, "Secure Authentication System for Web Application Based on a Public Key Cryptosystem.", Proceedings of Computer Security Symposium of IPSJ, 2C-3, 2014.
- Takuho, Mitsunaga, Ismail Omar, Yosuke Kuno, Naruhisa Tadokoro, Nobuaki Kondo, Hiroyuki Fujiki, Hiroshi Igarashi, "Implementation evaluation of attribute-based encryption with Key revocation function", Proceedings of the 73th National Convention of IPSJ, Vol.73, 2E-6, 2011.
- Yoshinori Matsumoto, Takuho Mitsunaga, Nobuaki Kondo, Yukio Rikisou, "A detection method using WAF to identify obfuscated malicious JavaScript", IEICE LOIS 2010-61 pp.99-104, 2011.

Book Chapters

- Japan Network Security Association. "*Protect your bussiness from*

cyber attacks(サイバー攻撃からビジネスを守る)", NTT Publishing, 2013.

- Information-technology Promotion Agency. "*Information Security White Paper* 2013(情報セキュリティ白書 2013)", IPA, 2013.

Other(Competitive research funding)

- R&D project on encrypted data sharing system with efficient key management tools for cloud computing, sponsored by Ministry of Economy, Trade and Industry, 2011-2012.

Acknowledgments

I would like to express my special appreciation and gratitude to all of those who were involved in this thesis, especially to my supervisors Professor Tatsuaki Okamoto and former Associate Professor Yoshifumi Manabe for their continuous and strong support for my research. I would also like to thank Professor Toru Ishida, Professor Yasuo Okabe and Professor Yoshimasa Nakamura for their assistance as my advisor committee members, and also to Associate Professor Shigeo Matsubara and Professor Tetsutaro Uehara at Ritsumeikan University for being my advisors during my Doctor's Program and providing some inspiring ideas and helpful advises for this thesis. My sincere thanks also goes to Associate Professor Masayuki Abe and the members of Professor Okamoto's Lab for their kind assistance, and finally to my colleagues Mr. Hiroshi Kobayashi and Ms. Yukako Uchida for helping me accomplish this research.

References

[1] "1password". Retrieved April 20, 2016, from https://agilebits.com/onepassword

[2] M. Abe and K. Suzuki, "M + 1st price auction using homomorphic encryption", Proceedings of Public Key Cryptography 2002, LNCS Vol. 2274, pp. 115-124, 2002.

[3] I. Abraham, D. Dolev, R. Gonen, and J. Y. Halpern, "Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation", Proceedings of 25th ACM Symp. Principles of Distributed Computing, pp. 53-62, 2006.

[4] I.Abraham. D. Dolev, and J. Y. Halpern. "Lower Bounds on Implementing Robust and Resilient Mediators". Theory of Cryptography Conference(TCC), LNCS Vol. 4948, pp. 302-319, 2008.

[5] R.Aumann. "Subjectivity and correlation in randomized strategies", Journal of Mathematical Economics, Vol. 1, No. 1, pp.67-96, 1974.

[6] J. Benaloh, "Secret sharing homomorphisms: keeping shares of a secret secret", Proceedings of CRYPTO 1986, LNCS Vol.263, pp. 251-260, 1986.

[7] D. Boneh and M. Franklin, "Efficient Generation of Shared RSA keys", Invited paper Public Key Cryptography 1998, LNCS Vol.1431, pp. 1-13, 1998.

[8] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts", Proceedings of Theory of Cryptography Conference 2005, LNCS Vol.3378, pp. 325-341, 2005.

[9] F. Boudot, B. Schoenmakers, J. Traor, "A Fair and Efficient Solution to the Socialist Millionaires' Problem.", Journal of Discrete Applied Mathematics Vol. 111(12), pp. 2336, 2000.

[10] Craig Gentry, "Fully homomorphic encryption using ideal lattices" Proceedings of the 41st annual ACM symposium on Theory of computing, pp. 169-178, May 2009.

[11] V. Chandola, A. Banerjee and V. Kumar. "Anomaly detection: A survey" Journal of ACM Computing Surveys(CSUR) , Vol. 41 Issue 3, pp.1-58, 2009.

[12] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", Communications of the ACM, pp 84-88, 1981.

[13] J.H. Cheon, J.S. Coron, J. Kim, M.S. Lee, T. Lepoint, M. Tibouchi, and A. Yun, "Batch Fully Homomorphic Encryption over the Integers", Proceeding of EUROCRYPT 2013, LNCS Vol. 7881, pp. 315335, 2013.

[14] E. Clarke, "Multipart pricing of public goods", Journal of Public Choice, Vol. 11(1), pp. 17-33, 1971.

[15] J.S. Coron, T. Lepoint, and M. Tibouchi, "Scale-invariant Fully Homomorphic Encryption over the Integers", Proceeding of PKC 2014, LNCS Vol. 8383, pp. 311328, 2014.

[16] R. Cramer, I. Damgard, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols", Proceedings of CRYPTO 1994, LNCS Vol. 839, pp. 174-187, 1994.

[17] "CryptoJS". Retrieved April 22, 2016, from https://code.google.com/p/crypto-js/

[18] D. Brown, "The exact security of ECDSA", Technical Report CORR 200-34, Retrieved April 20, 2016, from http://www.cacr.math.uwaterloo.ca

[19] M. Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan "Fully homomorphic encryption over the integers" Proceedings of EUROCRYPT 2010, LNCS Vol. 6110, pp. 24-43, 2010.

[20] Y. Dodis, S. Halevi, and T. Rabin "Cryptographic Solution to a Game Theoretic Problem", Proceedings of CRYPTO 2000, LNCS Vol. 1880, pp.112-130, 2000.

[21] B. Dodson, D. Sengupta, D. Boneh, M. S. Lam "Secure, Consumer-Friendly Web Authentication and Payments with a Phone", Proceedings of Mobile Computing, Applications, and Services, Vol. 76 of LNICST, pp. 17-38, 2010.

[22] D. Dolev and H. R. Strong "Authenticated algorithms for Byzantine agreement", SIAM Journal on Computing, Vol. 12(4), pp. 656-666, 1983.

[23] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", Proceedings of CRYPTO '84, LNCS Vol. 196, pp. 1018, 1985.

[24] R. Fagin, M. Naor, P. Wrinkler, “Comparing Information Without Leaking It”, Communications of the ACM Vol. 39(5), pp. 7785, 1996.

[25] “FIDO Alliance”, Retrieved April 22, 2016, from https://fidoalliance.org/

[26] M. K. Franklin and M. K. Reiter, “The design and implementation of a secure auction service”, IEEE Transactions on Software Engineering, Vol. 22, No. 5, pp. 302-312, 1995.

[27] S. Fujikawa, M. Yamauchi, H. Sunahara, “The proposal of the automatic login mechanism to the website which used the smart phone as the key(Japanese)”, Proceedings of IOTS symposium 2013, IPSJ, pp.53-57, 2013.

[28] O. Goldreich. “*Foundations of Cryptography vol.* 2 : *Basic Applications*” Cambridge University Press, 2004.

[29] O. Goldreich, S. Micali, and A. Wigderson, “How to play any mental game”, Proceedings of the 19th ACM Symposium on Theory of Computing, pp.218-229, 1987,.

[30] “Government Public Key Infrastructure”. Retrieved April 20, 2016, from http://www.gpki.go.jp/

[31] “GUIDE TO MEASURING THE INFORMATION SOCIETY 2009”. Retrieved April 20, 2016, from http://www.oecd.org/sti/sci-tech/43281062.pdf

[32] M. Harchol-Balter, T. Leighton, D. Lewin, “Resource discovery in distributed networks”, Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing, pp. 229-237, 1999.

[33] R. Hirano, M. Morii, “Secure and Effective Password Management System(Japanese)”, IEICE technical report. LOIS, vol. 111(286), pp.129-134, 2011.

[34] “How to Secure Your Web Site 5th Edition”. Retrieved April 20, 2016, from https://www.ipa.go.jp/files/000017318.pdf

[35] “How to Use SQL Calls to Secure Your Web Site”. Retrieved April 20, 2016, from https://www.ipa.go.jp/files/000017321.pdf

[36] “Japan Consumer Credit Association(Japanese)”. Retrieved April 20, 2016, from
http://www.jcredit.or.jp/information/statistics/download/
toukei_03_g_160331.pdf

[37] “JSEncrypt - Travis Tidwell”. Retrieved April 20, 2016, from http://travistidwell.com/jsencrypt/

[38] M.Jakobsson and A.Juels, “Mix and Match: Secure Function Evaluation via Ciphertexts”, Proceedings of Asiacrypt 2000, LNCS Vol. 1976, pp. 129-140, 2000.

[39] A. Juels and M. Szydlo, “A Two-Server Sealed-Bid Auction Protocol”, Proceedings of Financial Cryptography 2002, LNCS Vol. 2357, pp.72-86, 2002.

[40] J. Katz, “Bridging Game Theory and Cryptography:Recent Results and Future Directions”, Proceedings of Theory of Cryptography Conference 2008, LNCS Vol. 4948, pp. 251-272, 2008.

[41] T. Kerins, W.P. Marnane, E.M. Popovici, P.S.L.M. Barreto, “Hardware accelerators for pairing based cryptosystems”, IEE Proceedings Information Security 2005, Vol. 152, No1, pp.47- 56, 2005.

[42] T. Kobayashi, R. Yoshida and T. Yamamoto “Fine-grained access control system with public key(Japanese)”, The 31st Symposium on Cryptography and Information Security, SCIS 2014, 3C2-4, 2014.

[43] C. Kruegel and G. Vigna, “Anomaly detection of web-based attacks”, Proceedings of the 10th ACM conference on computer and communications security, pp. 251-261, 2003.

[44] K. Kurosawa and W. Ogata, “Bit-Slice Auction Circuit”, Proceedings of the 7th European Symposium on Research in Computer Security 2002, LNCS Vol.2502, pp. 24-38, 2002.

[45] L. Lamport, M. Fischer, “Byzantine generals and transactions commit protocols”, Technical Report Opus 62. SRI International, 1982.

[46] L. Lamport, R. Shostak, M. Pease, “The Byzantine Generals Problem” ACM Transactions on Programming Languages and Systems, Vol. 4(3), pp. 382-401, 1982.

[47] H. Lipmaa, N. Asokan, and V. Niemi. "Secure Vickrey auctions without threshold trust", Proceedings of the 6th Annual Conference on Financial Cryptography, LNCS Vol. 2357, pp. 87-101, 2002.

[48] M. Larson, C. Hu, R. Li, W. Li, X. Cheng "Secure Auctions without an Auctioneer via Verifiable Secret Sharing.", Proceedings of the 2015 Workshop on Privacy-Aware Mobile Computing, pp. 1-6, 2015.

[49] I. Loutfi, A. Jsang, "FIDO Trust Requirements, Proceedings of 20th Nordic Conference, LNCS Vol. 9417, pp 139-155, 2015.

[50] S. Matsubara, "Some Problems in an Allocation Mechanism for Digital Goods" IEICE technical report. IA, vol.101(535), pp.41-48, 2002.

[51] "Results of the E-Commerce Market Survey Compiled", Retrieved April 20, 2016, from http://www.meti.go.jp/english/press/2015/0529_02.html

[52] Commerce and Information Policy Bureau, METI(2011). "*Report on E − commerce Transaction* 2011". Tokyo:Research Institute of Economy, Trade and Industry Press.

[53] S. Matsubara and M. Yokoo, "Fraud-free Exchange Mechanisms in Electronic Commerce(Japanese)", Journal of Japanese Society for Artificial Intelligence, Vol.15, No.5, pp. 912-921, 2000.

[54] T. Mitsunaga, Y. Manabe, and T. Okamoto, "Efficient Secure Auction Protocols Based on the Boneh-Goh-Nissim Encryption", IEICE transactions on Fundamentals of Electronics, Communications and Computer Sciences Vol.E96-A, No.1, pp.68-75, 2013.

[55] T. Mitsunaga, Y. Manabe, and T. Okamoto, "A Secure M + 1st Price Auction Protocol Based on Bit Slice Circuits" Proceedings of the International Workshop on Security 2011, LNCS Vol.7038, pp. 51-64, 2011.

[56] J. Mirkovic, P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms", ACM SIGCOMM Computer Communication Review, Vol. 34 Issue 2, pp. 39-53, 2004.

[57] M. Naor, B. Pinkas, "Oblivious Transfer and Polynomial Evaluation", Proceedings of the 31st Annual ACM Symposium on the Theory of Computing, pp. 245-254, 1999.

[58] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and

mechanism design", Proceedings of the 1st ACM Conference on Electronic Commerce (ACM-EC), ACM press, pp.129-139, 1999.

[59] T. Nishide, M. Iwamoto, A. Iwasaki, and K. Ohta, "Secure (M+1)st-Price Auction with Automatic Tie-Break", Proceedings of 6th International Conference on Trustworthy Systems (Intrust), LNCS, Vol. 9473, pp. 422-437 , 2014.

[60] M. Nojoumian, D. R. Stinson, "Efficient Sealed-Bid Auction Protocols Using Verifiable Secret Sharing", Proceedings of ISPEC 2014, LNCS, Vol. 8434, pp. 302-31, 2014.

[61] "NPA Japan Countermeasure against Cybercrime(Japanese)". Retrieved April 20, 2016, from https://www.npa.go.jp/cyber/statics/index.html

[62] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring", Proceedings of Eurocrypt 1998, LNCS Vol. 1403, pp. 308-318, 1998.

[63] "OWASP Secure Coding Practices Quick Reference Guide". Retrieved April 20, 2016, from https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

[64] "OWASP Top 10 2013, Retrieved April 20, 2016". from http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20%202013.pdf

[65] P.Pallier, "Public-key cryptosystems based on composite degree residuosity classes", Proceedings of Eurocrypt 1999, LNCS Vol. 1592, pp. 223-238, 1999.

[66] C. Park, K. Itoh, and K.Kurosawa, "All/nothing election scheme and anonymous channel", Proceedings of Eurocrypt 1993, LNCS Vol. 765, pp. 248-259, 1993.

[67] T. Peng, C. Leckie, K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems", Journal of ACM Computing Surveys(CSUR) 2007, Vol. 39 Issue 1(3), pp.1-42, 2007.

[68] E. Rasmusen,
"*Games and Information : An Introduction to Game Theory*"
Blackwell Publishing 1994.

[69] "Request for Comments: 2898: PKCS #5: Password-Based Cryp-

tography Specification Version 2.0". Retrieved April 20, 2016, from https://www.ietf.org/rfc/rfc2898.txt

[70] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", Communication of the ACM, pp. 120-126, 1978.

[71] "RSA and ECC in JavaScript". Retrieved April 20, 2016, from http://www-cs-students.stanford.edu/ tjw/jsbn/'

[72] M. Shimaoka, T. Kataoka, S. Tanimoto, T. Nishimura, K. Yamaji, M. Nakamura, N. Sonehara, Y. Okabe, "Design of Architecture for University PKI" IEICE Transactions on Communication, Vol. J94-B, No. 10, pp. 1246-1260, 2011.

[73] Y. Tamura, T. Shiotsuki, and A. Miyaji, "Efficient Proxy-bidding system", IEICE Transactions on Fundamentals. Vol. J87-A, No.6, 835-842, 2004.

[74] "TrendLabs Annual Security Roundup(Japanese)". Retrieved April 20, 2016, from http://www.trendmicro.co.jp/cloud-content/jp/pdfs/security-intelligence/threat-report/pdf-2013asr-20140217.pdf?cm_sp=Corp-_-sr-_-2013asr

[75] "UPKI Digital certificate issuance service(Japanese)". Retrieved April 20, 2016, from https://certs.nii.ac.jp/

[76] W. Vickrey, "Counterspeculation, Auctions, and Competitive Sealed Tenders.", Journal of Finance, Vol. 16(1), pp. 8-37, 1961.

[77] A. Vapen, N. Shahmehri, "Security Levels for Web Authentication Using Mobile Phones", Proceedings of Privacy and Identity Management for Life 2010, pp. 130-143, 2010.

[78] A. Yao, "Protocols for secure computations", Proceedings of IEEE Symposium on Foundations of Computer Science, pp.160-164, 1982.

[79] N. Yildirim and A. Varol, "Android based mobile application development for web login authentication using fingerprint recognition feature", Proceedings of IEEE Signal Processing and Communications Applications Conference (SIU), pp. 2662-2665 , 2015.

[80] Makoto Yokoo(2006). "*fundamental of auction theory*(Japanese)". Tokyo:Tokyo Denki University Press.

[81] "Web Storage - World Wide Web Consortium". Retrieved April 20, 2016,

from http://www.w3.org/TR/webstorage/