DISSERTATION

A Study on the Secure Online Examination System

Graduate School of

Natural Science & Technology

Kanazawa University

Division of Electrical Engineering and Computer Science

Student Number: 1323112010

Name : **Abdul Wahid**

Chief advisor : **Prof. Masahiro MAMBO**

July 1, 2016

Acknowledgements

First of all, I would like to express my gratitude to my academic supervisor, Professor Masahiro Mambo, for his guidance, suggestion and feedback. During the three year doctoral course taught me how to do research and to write paper, which have supported in achieving my academic goal of getting a Ph.D. in information security. Thank you very much for your time, effort, knowledge, motivation and dedication during supervising me. I wish to express grateful acknowledgement to Professor Yasushi Sengoku and Professor Kenji Yasunaga for the discussion and comments. I would also like to thank for staffs of Graduate School of Natural Science and Technology, Kanazawa University for their kind help and care during the years I was studying here.

I am deepest thankful for my parents (H.M.Yunus-Hj.Nuhuriah) and my parents in law (H.Hairil Muin-Hj.Rahmawati) for supporting, praying, paying and motivating me in my study and life. My wife, Dian Herawati Hairil, my daughters, Alya and Aisyah, my sister, Tetty Wahyuni, thank for supporting and praying for me. Without their boundless love, endless patience, encouragement and sacrifice, I would not accomplish my study. I am thankful to all members of ISec laboratory for their valuable helps, especially for Mr. Takahashi, Mr. Nakamura and Mr. Kosugi for discussing, sharing knowledge and helping me for my daily activities here. All members of Indonesian Student Assosiations in Kanazawa, thank you very much for all of your support and discussion.

I am also grateful to Indonesian Directorate Higher Education Ministry of Education and Culture (DIKTI), Indonesia, and Kanazawa University, Japan, for fellowship and to Universitas Negeri Makassar for allowing me to continue my Ph.D. study.

Finally, the last but not least, I would like to thank for the deepest heart to my God ALLAH S.W.T. Without ALLAH SWT, I can not do everything. Thank you everything,

Abdul Wahid
Kanazawa, June 2016

Abstract

Implementation of secure online examination system has been a hot topic in the educational world in the last decade. Issues that should be addressed in the secure online examination system are computer and network security issues of the systems and prevention of cheating by participants. In our research, we provide a website application and a secure network design which prevents cheating by any participant among examinee, administrator, and examiner. Different security features of the online examination system are discussed both from the website application aspect and network design aspect.

Unfortunately, website application and network design cannot meet some security requirements because of several inside and outside attacks and malicious behaviors of bribed, corrupted or unfair examiners and untrusted exam authority, and we construct a particular online examination protocol to prevent them.

We design an online examination protocol based on certificateless signcryption and prove their security properties under the formal analysis using ProVerif software. The proposed online examination protocol has several advantages over existing protocols such that there is no certificate unlike public key infrastructure, no key escrow and lower computational cost by virtue of the signcryption scheme.

Our results show that some of OES problems both of data security issue such that scanning port attack and cheating problem especially by examinee can be handled over the web application and network design system. While some others will be handled by particular OES protocol. ProVerif shows that our proposed protocol is secure under some privacy and authentication properties.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The examination is one way to measure the success of learning process or obtaining qualified human resources. In the field of training, the exam is intended to measure the level of achievement by students or learners, so that we can determine the level of understandings of the study being taken. In the context of the recruitment of new employees, the exam is intended to obtain qualified human resources [1], [2].

All of the examination systems including the national exam system in all levels of education, whether it is an exam for students or exams for teachers, have begun to shift from the manual or paper-based exam system to electronic exam systems in order to make it more practical and effective. According to the resources utilization, electronic exam can be categorized into three types that is:

1. Computer-assisted examination; This type is not fully use computer. Computer is only used to support the exam, i.e. exam uses computer only for showing the exam questions or only for marking the exam results.

2. Computer-based examination; Here, exam is taken on computer.

3. Online examination; Here, besides exam is taken on computer, exam requires Internet connection to distribute the questions, answers and results of exam. Online exam supports long distance or remote exam.

Nowadays, online examination systems becomes a hot topic. This type of examination system is computerized, in which examinees answer test questions through a computer. Assessment is conducted directly by the system, and examinees will receive their results immediately after the exam [3]. Several researches and applications with any features have proposed to implement it.

Although the online examination system has its advantages, computerization incurs security problems. Each exam sessions need to deal with cheating that could occur. So far, online examination system has mostly focused on system security itself, such as the design of access control, defense against attacks, closing security holes in the application such as PHP, SQL and operating system or applications of encryption encryptions to database and communication. However, there are a variety of cheating methods more crucial in online examination systems than in conventional exam systems.

Cheating usually exploits weaknesses in the implementation of conventional and online exams. Along with the development of information technology, there is also an increase in more diverse and sophisticated cheating methods. An example is the use of spy cameras or modern communication tools that are modified to make it undetectable by the exam committee.

1.1 Aims and Objectives

This research aims to study the problems in Online Examination System which has described above, especially those which still has not been considered in the previous research. We intend to achieve those aims through four objectives.

1. **To identify problems in Online Examination systems (OES)**. This is a very fundamental objective as it provides the basis of further research and determines the model of framework and protocol of OES that will be developed.

2. **To develop a basic framework of OES**. This is an important objective to achieve a solution in OES problems. We will construct web-based application OES and design network system which solves some problems which have been identified.

3. **To design a new fast and secure protocol for developed framework of OES**.This objective consists in proposing a new faster and effective exam protocols that meet the security requirements which satisfies to our application and network design. It requires combining secure cryptographic schemes to guarantee the often contrasting requirements. This protocol is expected to be a future security protocol, especially for OES.

4. **To evaluate, security aspects of the designed OES protocol. To this end, we evaluate the designed protocol under the computational model and formal model**. This objective is to expand

the formal model analysis of OES protocol by considering also the user. The desired outcome is to understand how user's choices may influence the security of exam protocols.

1.2 Contributions

This dissertation addresses the four objectives outlined in the previous section. The dominant aim of this work is to construct a secure Online Examination System which secures in both of network system and several common cheating methods. We claim the following issues as the contributions derived from our work:

1. We have constructed a basic framework of secure online examination system which can prevent several network penetration attacks and common cheating methods without a special browser and e-monitoring system.

2. We have designed a secure and efficient communication protocol using certificateless signcryption method.

1.3 Outline of the Dissertation

This dissertation is structured in five chapters. Most of the contents of the dissertation have been published in conference papers or submitted to journal articles. In the following, we outline the contents of each chapter.

1. Chapter 1: In this chapter, we have discussed the introduction to OES, aims, objectives and contributions of our research.

2. Chapter 2: In this chapter, we present the preliminaries where are required for implementation of our proposed scheme. Here, we will discuss mathematic of cryptography, elliptic curve cryptography, security protocols and a little describing Proverif tools.

3. Chapter 3: Here we will discuss the problem of Online Examination System, several related works and then our proposed model and implementation. In the end of this chapter, we will analyze our features.

4. Chapter 4: Here we have proposed a Certificateless Signcryption (CLSC) protocol scheme and try to analyze the security and compared their complexities and efficiency with another scheme.

5. Chapter 5: In this chapter, we describe our OES protocol based on Certificateless Signcryption which has been proposed in the previous chapter. Here, we will explain the formal analysis for proving the security of this proposed EOS protocol.

6. Chapter 6: The last chapter, we present conclusion our research and future work.

Chapter 2

Preliminaries

2.1 Mathematic of Cryptography

In this section, we will be going to explain various mathematical properties of cryptography that is useful to understand the mathematics description of this dissertation. Some important function like group, ring, field, elliptic curve will be discussed here.

2.1.1 Modular Arithmetic

Modular arithmetic is defined as a system of arithmetic for integers, where we are interested in the only remainder, not quotient [4].
Set of Residues: Z_n Here Z is the set of an integer. Modulo operations result always gives a non-negative integer. Suppose n is the modulo operation then the value of n is between 0 to $n-1$, suppose a mod n is any modulo operation where $'a'$ is any integer then result varies between 0 to $n-1$.
As example: $Z_n = \{0, 1, 2, \cdots, (n-1)\}$
$Z_2 = \{0, 1\}$
$Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$Z_{11} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
Additive Inverse: Suppose x and y are two number in Z_n then it is called additive inverse of one another if $x + y = 0(mod\ n)$
As an example: in $Z_{14}, 14 - 4 = 10$ is additive inverse of 4, so in generalized way for $Z_n, y = n - x$
Multiplicative Inverse: If there are two numbers x and y which are multiplicative inverse of each other. If $x \times y \equiv 1(mod\ n)$ in Z_{10}, the multiplicative inverse of 3 is 7 because $3 \times 7 \equiv 1(mod\ 10)$. The integer x in Z_n has a multiplicative inverse exist only if $\gcd(n, x) = 1$.

For example, 8 have no multiplicative inverse in Z_{10} because $\gcd(10, 8) \neq 1$.
The Set Z_n: Its 3 instances are shown below:

1. Z_n^*: The set, Z_n^* is defined as a subset of Z_n and it contains elements of set Z_n that have a multiplicative inverse. In the set Z_n, all the elements have an additive inverse, but only some members have a multiplicative inverse.
Example:$Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $Z_{10}^* = \{1, 3, 7, 9\}$

2. Z_p: In the set Z_p, p is a prime number and same as Z_n i.e., contains all elements from 0 to $p - 1$. In Z_p, all the elements. Note: We need to use Z_n when additive inverses are needed; we need to use Z_n^* when multiplicative inverses.
Example:$Z_{13} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

3. Z_p^*: In the Set Z_p^*, p is a prime number and same as Z_n^* i.e., contains all the elements from 1 to $p - 1$. In Z_p^* , all the elements have additive and multiplicative inverse.
Example:$Z_{13}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

2.1.2 Algebraic Structures

In this session, we briefly discuss the subject of algebraic structure. An algebraic structure is defined as the set of the element with an operation that is applied to the element of the set. There are three common algebraic structures known as groups, rings, and fields [4].

1. **Groups**: A group (G) can be defined as a set of elements, which satisfies the following four properties with a binary operation, denoted as $G = \langle\{\cdots\}, \cdot\rangle$.

   - Closure: We can define as, If $a, b \in G$, then $c = a\cdot b \in G$.
   - Associativity: We can define as, If $a, b, c \in G$ then $(a\cdot b)\cdot c = a\cdot (b\cdot c)$.
   - Identity: We can define as, $\forall a \in G$, there exists an identity element e, such that $e\cdot a = a\cdot e = a$.
   - Inverse: We can define as, $\forall a \in G, \exists \bar{a}$, called the inverse of a, such that $a\cdot \bar{a} = \bar{a}\cdot a = e$.

   If there is a group which satisfies above four properties along with commutative property, it is called commutative group or Abelian group. Commutative property means $\forall a \in G$, we have $a\cdot b = b\cdot a$.

Finite Group: A set can be called as finite group if it contains a finite number of elements, otherwise it is an infinite group.

Order of a Group: Order of a group is total number of elements that contains in a group, i.e., $|G|$.

Subgroups: A subgroup is a subset of group and subgroup itself is a group. If G and H are two groups of the same operation and elements of H is a subset of element of G, then H is subgroup of G. The above definition implies that:

- If $a, b \in G$ and H, then $c = a \cdot b \in G$ and H.
- Both group and subgroup share the same identity element.
- If $a \in G$ and H, then $\bar{a} \in G$ and H.
- The group made of identity element of $G, H = \langle \{e\}, \cdot \rangle$ is a subgroup of G.
- Each group is a subgroup of itself.

Cyclic Subgroups: A subgroup of a group is called cyclic subgroup if all the elements of the group generated using the power of an element. The term power means repeatedly applying the group operation to the element.
$a^n \rightarrow a \cdot a \cdot a \cdot \cdots \cdot a$ (n times).

Cyclic Groups: The element that generates all the elements of the cyclic subgroup can also generate all the elements group is called a generator. A cyclic group is a group that itself own cyclic subgroup. If g is a generator, the element in the finite group can be written as $\{e, g, g^2, \cdots, g^{n-1}\}$,where $g^n = e$.

2. **Rings**: A Ring(R) is a set of two binary operations. It is denoted as $R = \langle \{\cdots\}, \cdot, + \rangle$. The first and second operation must satisfy all five and two properties respectively. In addition, the second operation must be distributed over first, means that for all a, b and c elements of R. We have $a + (b \cdot c) = (a + b) \cdot (a + c)$ and $(a \cdot b) + c = (a + c) \cdot (b + c)$. If the second operation satisfies commutative operation, then the ring is called commutative ring.

3. **Fields**: A Field (F) is a set of elements with a binary operation, denoted as $F = \langle \{\cdots\}, \cdot, + \rangle$. Both two operations satisfy all five properties except the identity of the first operation has no inverse.

2.2 Elliptic Curve Cryptography

Elliptic-curve system in cryptography is suggested in 1985 [5] by Victor Miller and Neal Koblitz as an alternative mechanism for implementing public-key cryptography based on an elliptic curve over a finite field. ECC is based on discrete logarithm that is much more difficult to challenge at equivalent key lengths as compared to other public key cryptography. ECC will use the smaller key if we compare to other public key cryptography in the same security level. So, it is used widely in lower resource system like mobile communication.

Definition : An Elliptic-curve [5] over a field which is finite, is a non-singular cubic curve that has 2 variables, where $f(P, Q) = 0$. The field P is usually taken to be the complex numbers, real numbers, rational numbers, algebraic expressions of rational numbers or a finite field. By, non-singular means all 3 roots of EC must be distinct.

General form of elliptic-curve (EC):

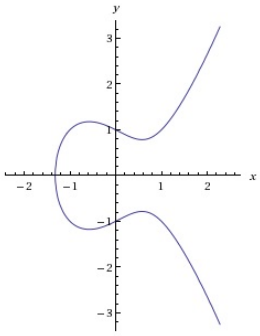


Figure 2.1: graphical representation of elliptic curve $y^2 = x^3 - x + 1$

Properties:
1. Symmetric over x-axis.
2. The cubic curve in the variable x.

Any elliptic curve can be defined by following equation. $A^2 = B^3 + aB + b$, here B is not a continuous point, chosen from particular field $GF(P)$ or $GF(2^k)$. The figure 2.1 shows the elliptic curve of equation $y^2 = x^3 - x + 1$.

The advantages of ECC:

The National Institute of Standards and Technology recommended the key sizes to protect keys used in conventional encryption algorithms like the (DES) and (AES) together with the key sizes for RSA, Diffie-Hellman and elliptic curves that are needed to provide equivalent security are given in Table 2.1.

Table 2.1: Key sizes foe equivalent security levels (in bits) [6]

Symmetric	ECC	DH/DSA/RSA
80	160	1024
112	185	2024
120	237	2560
128	256	3072
256	512	15360

From above table, we can see that if the symmetric key size increases the required key sizes for RSA and Diffie-Hellman increase at a much faster rate than the required key sizes for elliptic curve cryptosystems. Hence, elliptic curve systems offer more security per bit increase in key size than either RSA or Diffie-Hellman public key systems.

Elliptic Curve Hardest Problem

This section describes the definition of the hard computational problems in which the security of the proposed scheme relies on [7].

1. Elliptic Curve Discrete Logarithm Problem
   With the given two point of an elliptic curve A and B, where $A = k \cdot B$, it is difficult to find out the value of k.

2. Elliptic Curve Diffie-Hellman Problem
   With the given two points of elliptic curve A and B, where $A = c \cdot G$ and $B = d \cdot G$ without c and d, it is difficult to find out another point $K = c \cdot d \cdot G$. The ECDLP and ECDHP are computationally infeasible problems.

2.3 Security Protocols

Security protocol is one of the most important mechanisms in providing security networks because crucial data or information is hidden by this mechanism. Some security protocols are built for a specific use with a variety of purposes, such as secure channel (SSH/SSL or TLS/IPSec), wi-fi (WEP/ WPA/ WPA2), banking, e-voting, certified mail, mobile phone, etc. Figure 2.2 illustrates one example of a protocol, namely the Denning-Sacco key distribution protocol [8]. The goal of this protocol is for the key k to be a secret key shared between A and B, so that s can be kept confidential upon delivery because it is encrypted by using a key k.

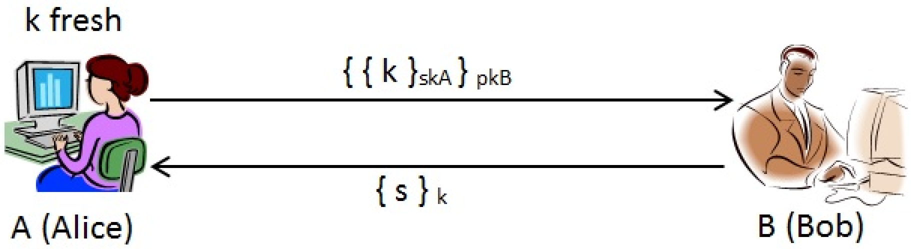


Figure 2.2: Simplified Denning-Sacco key distribution Protocol

In reality, this protocol still cannot be considered secure, for an active C attacker could impersonate A and obtain the secret s. Figure 2.3 illustrates how the (well-known) attack is against this protocol.

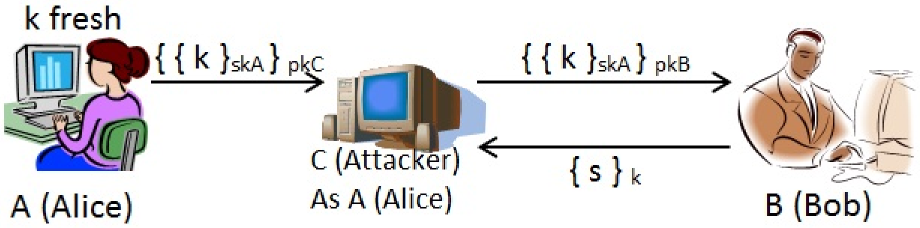


Figure 2.3: Attacking of Denning-Sacco key distribution Protocol

With a little analysis and modifications to this protocol, C cannot impersonate A as shown in the Figure 2.4. This is because in the previous attack, the first message namely $\{\{A, C, k\}_{skA}\}_{pkB}$,cannot be accepted by B.

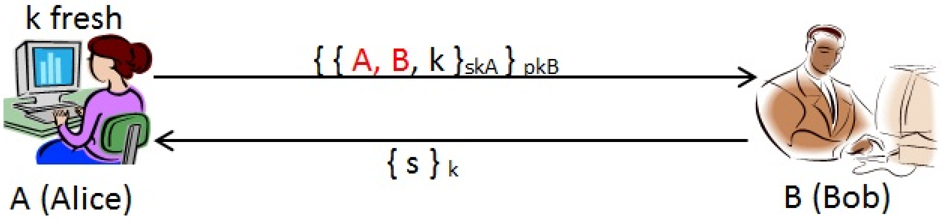


Figure 2.4: Revision of Denning-Sacco key distribution Protocol

In order to analyzing and proofing security level of protocol, cryptographer needs to verify the protocols. Now, the verification of protocols has been and still a very active research topic because the protocol design is error prone and these errors are not detected by testing, they appear only in the presence of an adversary. Besides, errors can have serious consequences. There are two main frameworks for analyzing security protocols:

1. The Dolev - Yao model: a formal or abstract model. Here, the cryptographic primitives are ideal black boxes and adversary or attacker uses only those primitives. Proof can be done automatically by Pro Verif tool [9].

2. The Computational model: a realistic model which the cryptography primitives are functions on bit-strings and the adversary is a polynomial-time Turing machine. Several proofs are done manually but some automatic prover sound in the computational model. One of them is Crypto Verif [10].

Both of models usually assume active attackers which have some ability, e.g. the attacker can intercept all messages sent through the network, compute messages and send messages through the network.

The Dolev-Yao Model for OES Protocol: The threat model of an online examination protocol consist of a Dolev-Yao attacker who has full control of the network, namely of the public channel. In the public channels, an attacker can eavesdrop, drop, substitute, duplicate, and delay messages that senders sent to receivers. In addition the ability of an attacker can be extended with corrupted principals. He can also inject message of his choice into the public channel, and exploid the algebraic properties of cryptographic primitives because a theory of equational. However, We has private channel. In private channel, attackers has not control which are normally used to model out-of-band communication between processes. The attacker cannot see and know all communications happen over private channel.

2.4 ProVerif

ProVerif is a tool created specifically to verify cryptographic protocol [9]. ProVerif has the ability to verify the protocol with an unlimited number of sessions. A model that is run on a limited number of sessions has the disadvantage of if the user has specified the number of sessions and the results issued by the tool indicates that there was no attack on the protocol according to the number of sessions stated. Thus, the existence of attacks in a larger number of sessions cannot be ascertained. ProVerif allows users to find the attack that occurred by using a representation with an unlimited number of sessions.

At first, ProVerif can only accept modeling input that is written using Horn clause representation. With this representation, ProVerif can verify security secrecy objectives explicitly and authenticate implicitly. ProVerif was then developed in order to receive modeling input written in the calculus model representation, as well as to verify the authentication security objectives explicitly.

When ProVerif claims that a protocol does meet the security secrecy objectives and/or authentication, the secrecy and/or authentication of the protocol is definitely guaranteed. In other words, no attacks were found. However, if ProVerif issues an output of a possible attack channel, then the output should still be examined. The results of this output can be either an actual attack channel or only a false attack.

Process calculus is an approach used to model processes that run in parallel. ProVerif uses a representation of pi calculus enriched with some syntax (Table 2.2) to model the cryptographic protocol and security objectives. This representation is hereinafter referred to as the representations process calculus.

Table 2.2: Syntax of Process Calculus

$M, N ::=$	*terms*
x, y, z	*variable*
a, b, c, k	*name*
$\mathtt{f}(M_1, \cdots, M_n)$	*constructor_application*
$P, Q ::=$	*Processes*
$\bar{M}\langle N\rangle \cdot P$	*output*
$M(x) \cdot P$	*input*
$\mathtt{0}$	*nil*
$P \vert Q$	*parallel_composition*
$!P$	*replication*
$(va)P$	*restriction*
$\mathtt{let}\ x = g(M_1, \cdots, M_n)\ \mathtt{in}\ P\ \mathtt{else}\ Q$	*destructor_application*
Q	*event*
$begin(M) \cdot P$	*begin_event*
$end(M) \cdot P$	*end_event*
$begin\_ex(M)$	*executed_begin_event*
$end\_ex(M)$	*executed_end_event*

The syntax of process calculus consists of definitions of terms and process. Terms could be a variable, name, or constructor application. While the process can be:

- Process output $M\langle N\rangle\cdot P$ means sending message N to channel M and then executing process P.

- Process input $M(x)\cdot P$ means receiving message x from channel M and then executing process P. In executing process P, x in P will be substituted with a message received through M.

- Process *0* (zero) will not do anything.

- Process $P|Q$ is a parallel composition of P and Q.

- Replication $!P$ represents a number of process P copies that are unlimited in number and run parallel, namely $P|P|P\cdots$.

- Restriction $(va)P$ form a new name a and then executes process P.

- A process that is a destructor application have the form of let $x = g(M_1, \cdots, M_n)$ in P else Q. This process means it will evaluate destructor $x = g(M_1, \cdots, M_n)$. If evaluation is successful, then the evaluation result becomes the value of evaluation x and process P is then executed.

Process $\mathit{begin}(M)\cdot P$, $\mathit{end}(M)\cdot P$, $\mathit{begin\_ex}(M)$, and $\mathit{end\_ex}(M)$ is typically used in specifying authentication. Process $\mathit{begin}(M)\cdot P$ issues event $\mathit{begin}(M)$ and then executes process P. This process is used as a sign that the protocol participant has started its role in a protocol session. This process is paired with the $\mathit{end}(M)\cdot P$ process which means issuing event $\mathit{end}(M)$ and then executing P. This process is used as a sign that the protocol participants have terminated its role in a protocol session. $\mathit{begin\_ex}(M)$ and $\mathit{end\_ex}(M)$ processes are used to remember that event $\mathit{begin}(M)$ and $\mathit{end}(M)$ has been executed. Both of these processes are not used directly in the modeling protocol but in the reduction process that defines the semantics of the process calculus being used.

In addition to the definition of the term process above, there is also the definition of *let* $x = M$ in P that gives the same meaning to run process P in which the emergence of x in P will be substituted with M. Additionally, the definition *if* $M = N$ then P else Q gives the same meaning as *let* $x = \mathit{equal}\ (M, N)$ in P else *equal* Q. Equal destructor has a definition of *equal* $(M, N) \rightarrow M$. The constructor and destructor for the cryptography operations can be found in table 2.3.

As an example, a simplified Needham-Schoeder Public Key protocol is used with the following specifications:
1. $A \rightarrow B : \{NA\}_{pkB}$

Table 2.3: Constructor and Destructor in Process Calculus

	Symmetric key encryption
`Constructor`	Encrypt message M with key N, $senc(M,N)$
`Destructor`	Decrypt $sdec(senc(M,N),N) \rightarrow M$
	Asymmetric key encryption
`Constructor`	Generate public key from private $N, pk(N)$
	Encrypt message M with public key N, $penc(M,N)$
`Destructor`	Decrypt $pdec(penc(M,pk(N)),N) \rightarrow M$
	Digital Signature
`Constructor`	Signature message M with private key N, $sign(M,N)$
`Destructor`	Verify signature $checksign(sign(M,N),pk(N) \rightarrow M$
	Message without signature $getmess(sign(M,N) \rightarrow M$
	Hash Function
`Constructor`	Hash function of message M, $H(M)$
`Destructor`	-
	Tuple with n arity
`Constructor`	Tuple $ntuple(M_1,\cdots,M_n)$
`Destructor`	Projection, $i_n^{th}(ntuple(M_1,\cdots,M_n)) \rightarrow M_i, i \in (1,\cdots,n)$

2. $B \rightarrow A : \{NA, NB\}_{pkA}$
3. $A \rightarrow B : \{NB\}_{pkB}$

The protocol can be represented with process calculus as seen in the following process P:

$$\begin{aligned} P = \quad & (vskA)(vskB)letpkA = pk(skA)inletpkB = pk(skB)in \\ & cpkA.cpkB.((!PA(skA,pkA))(!PB(skB,pkB,pkA))) \end{aligned}$$

With P_A dan P_B defined as follows:

$$\begin{aligned} P_A(skA,pkA) = \quad & c(x\_pkB).begin(x\_pkB).(v.Na)c\langle penc(Na,x\_pkB)\rangle.c(m_2). \\ & \text{let } w = pdec(m_2,skA) \text{ in let } na = 1_2^{st}(w)in \\ & \text{let } nb = 2_2^{nd}(w) \text{ in if } na = Na \text{ then} \\ & c\langle penc(Nb,x_pkB)\rangle.0 \end{aligned}$$

$$\begin{aligned} P_B(skB,pkB,pkA) = \quad & c(m_1).\text{ let } y\_Na = pdec(m_1,skB) \text{ in} \\ & \text{let } (vNb)c\langle penc(2tuple(y_Na,Nb),pkA)\rangle.c(m_3). \\ & \text{let } y\_Nb = pdec(m_3.skB) \text{ in if } y\_Nb = Nb \text{ then} \\ & end(pkB).0 \end{aligned}$$

Channel c has access to the public, including to the attacker. According to the Dolev-Yao model, the attacker can find out all the messages exchanged in this channel, create a new message from this information, and then send it to the same channel.

Process P begins with the creation of private and public keys of A and B. The public key is then sent via channel c to model the public key that is made known to the public and became the initial information of the attacker. After that processes P_A and P_B will be run in parallel where each process is executed with an unlimited number of sessions.

Process P_A represents messages received and sent by A. In this process, A first receive the public key via channel c to indicate with whom A communicates. A then issues event $begin(x\_pkB)$ as a sign that it had started a session with parties that have x_pkB. A then creates nonce Na, encrypts it with the public key that it received and sends it through channel c. A later receives message m_2 which it decrypts with its private key and obtains two nonce, namely na and nb. If nonce na is the same as nonce Na previously sent through channel c, then A will be confident that it communicated with the owner of x_pkB and send the last message in the protocol, namely nonce nb (which it previously received) encrypted with public key x_pkB.

Meanwhile process P_B represents messages received and sent by B. In this process, B receives message m_1 that corresponds to a message sent by A, namely Na encrypted with public-key B. It then encrypts this message with its private-key and obtains nonce y_Na. Next, it creates nonce Nb and sends the nonce and y_Na encrypted with public-key A. B then receives message m_3 which corresponds to the message sent by A, namely Nb encrypted with public-key B. It then decrypts it with its private-key and obtains nonce y_Nb. If y_Nb is the same as the previous nonce Nb, it is sent through channel c then, B is convinced that it is communicating with A. It then issues event $end(pkB)$ as a sign that it has completed a session with A.

In applied phi-calculus, secrecy can be modelled as a reachability property. The secrecy of a term m is preserved if an attacker, defined as arbitrary process, cannot construct m from any run in of protocol. There are two definitions to model secrecy, name-distinct and reachability-based secrecy. A name-distinct process signifies that the name mentioned in a term appear unambiguously in the process either free or bound names. While reachability-based secrecy says that an attacker cannot build a process A that can output the secret term m.

In the other hand, the notion of observational equivalence can capture privacy requirements. Informally, two processes are observational equivalence if an observer cannot distinguish the process despite they might handle different data or perform distinct computation.

Authentication can be defined using correspondence assertions. An event e is a message emitted into a special channel that is not under the control of attacker. Event may contain arguments $M_1, , M_n$, which are never revealed to the attacker . Events do not change the behavior of process in which they are located, but normally flag important steps in the execution of protocol. To model correspondence assertions, we annotate processes with events such as $e\langle M_1, , M_n\rangle$ and reason about the relationship ($\rightarrow$) between events and their arguments in the form if an event $e\langle M_1, , M_n\rangle$ has been executed, then event $e\langle M_1, , M_n\rangle$ has been previously executed, which formalized as the following form:

$$e\langle M_1, , M_n\rangle \rightarrow e\langle M_1, , M_n\rangle$$

By adding key word inj, it is possible to model injective correspondence assertions, which signifies that if an event $e\langle M_1, , M_n\rangle$ has been executed, then a distinct earlier occurrence of event $e\langle M_1, , M_n\rangle$ has been previously executed. we formalized as the following form:

$$e\langle M_1, , M_n\rangle \rightarrow inje\langle M_1, , M_n\rangle$$

Chapter 3

Online Examination Systems

Online examination system (OES) is an exam based on the Internet without paper, each action of examination is conducted through the network e.g. the delivery of question sheets and answers to the test. In recently years, OES are really a research challenge for any situation of exam related with long distance learning and truly online. The environment situation of online examination cannot be totally controlled or can be fully controlled depend on the situations during an exam. We have to consider which situation of our online exam because the situation of exam will influence how to keep the security of our system and how to prevent cheating during the exam.

In order to make easier for design our system, we consider and make several assumptions as a target situation of our OES:

1. In the OES framework, there is a basic computer used by each participant. A large number of participants located in several places take exam at a fixed time and at a fixed axam seat.Limited number of supervisors are in each room during the exams.

2. OES consists of 3 entities which are the examinee, administrator and examiner. Each of this entity has a privileged access to different pages.

3. Examinees take the exam in a secure place or room such as a computer lab or ICT center which has already been set and registered for OES.

4. The examiner executes set-up exam questions from registered place or computer.

5. Whether grading process can be done automatically by the system or manually by the examiner depends on their type of questions.

6. Manual grading will be performed by examiner in a registered place.

3.1 Problem on OES

Before we go to the next step, we need to identify the problem of our OES related to the target situation above. As we know, OES cannot separate with computer network systems and the main problem here is security. Computers and network security problems occur due to the presence of securities hole in the system both of its network design and program coding. The existence of security hole allows someone both of inside attacker and outside attackers to access the system by illegally stealing exam questions and answers, making changes to existing value, or another type of modifications. Web design security is very important because it has a content that must be protected. Without any prevention method, anyone can penetrate into the web and obtain data stored on the web. There are several security aspects that should be guaranteed in the OES:

1. Database secrecy; There are so many sensitive data in OES that should be kept secret.

2. Data Integrity; We have to ensure that all received data during communication is real data.

3. Authenticity; Authenticities of all messages, transactions or other exchange of information before, during and after the exam must be ensured.

4. Data secrecy of transmission; OES is often constructed as a server and client system. Sensitive data transmitted among them should be protected.

5. Data access control; Data on OES only should be accessed on the specific time and place.

In addition to the security issue of the computer and networking systems, the other important issue in the OES is cheating prevention. There are many techniques that are often used by the examinee to obtain exam answers illegally, for instance, browsing the Internet, using messenger communication or other common cheating techniques. The following are some of the basic techniques used to get answers illegally during the exam:

1. Browsing on the Internet; Examinee can seek answers to questions by utilizing existing search engines like Google or Yahoo.

2. Using the Internet messenger for communication; Examinee can have discussions with others either existing in the same network or outside networks by using the messenger facility.

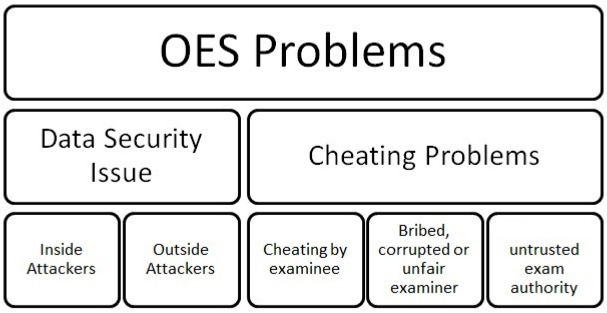


Figure 3.1: Problems of Online Examination System

3. Communication with others; Without or limited proctor supervision, the examinee can have discussions with other examinees in the same room or at a far distance via a portable device brought in by the examinee.

4. Access to local or external storage; Examinee can find an answer to accessing files on local or external storage such as flash drive or CD/DVD installed on the PC where they are doing the test.

5. Reading a book or tutorial directly; Examinee can find the answer by reading a book or tutorial directly because there is no exam supervisor.

6. Collusion; Examinee and administrator or examiner can have collude in order to increase their exam score.

Besides the cheating by examinee, there are some threats possible occurs in online examination system such that bribed, corrupted or unfair examiner and dishonest or untrusted exam authority. Figure 3.1 shows the problems diagram of online examination systems.

3.2 Related Work

According to [11], the basics dilemma in Online Examination system is integrity and secrecy of the questions, answers, grades and examines itself, collaborating and cheating examinees. These ”honesty control issues” also apply to traditional classroom courses in which the instructor uses online, out-of-class exams to save classroom time for non-exam purposes. They propose eight control procedure to solve this dilemma, they are an exam should be scheduled for a specific date and time, an exam should close when the allotted time period for work expires, an exam should be open to Internet access for only a limited time period, examinees can only solve one question at a time and cannot access completed questions. A student can access the exam only once, an exam should be limited to special purpose browser, an exam should be a randomizing question and answer choices, and about one-third of objective type questions should be rotated or modified in each exam every term.

There are several Content Management System (CMS) for online education that utilizes web-based commercial courses management software such as Moodle, Web CT, Blackboard, or software developed in-house. This software is not used widely for online exams, due to security vulnerabilities, and the system must rely on students honesty or their having an honour code [1]. Besides, CMS still does not have enough cheating control in their system. In order to prevent cheating by examinee, many general CMSs have functions to form test-problem sets randomly from pools and to analyze answers from students. For example of Blackboard, one of the famous CMSs, can compose a test which poses a different set of problems to every student based on a pool and categories of problems, and inform its examiner of statistics such as difficulties and discrimination of the problems by analyzing student answers based on the item responses theory.

A simple solution to the issue of computer and network security and cheating on online exams was proposed in many papers [1, 2, 3, 12, 13, 14]. One solution is called an enhanced Security Control System in the Online Exam (SeCOnE) which is based on group cryptography with an e-monitoring scheme [13]. The other cryptographic schemes was proposed [1, 2, 12, 15]. The control of cheating problems in online examination system by using camera as e-monitoring was proposed in [1, 2, 15].

Implementation of online examination system has been offered in [16, 17]. They tried to build OES by offering some important features in their models such as the user's registration, examination instruction, a valid time of exam and time reminders. They used three entities that will access to their system namely admin, teacher, and student.

On the other hand, the characteristics and potential ways of cheating during the online exam process and the shortage of existing Online Examination Systems, anti-cheating measures were analyzed by [14]. They provide two solutions to prevent cheating. The first solution is based on the automatic generating examination paper algorithm which takes advantage of the knapsack problem principle. The other solution is based on the self-developed ActiveX control.

Another thing that has not been considered in the existing system is the possibility of collusion between the examinees with examiner or administrator to improve the exam results. Or cheating can be committed by the examiner or administrator to change the results of an exam.

After reviewing several working above, we found some existing problems which will be considered in our discussion as follow:

1. E-monitoring as a solution for cheating problem needs high cost and widely bandwidth, so we try to reduce some examinee cheating techniques by another method without e-monitoring.

2. In order to prevent cheating by examinee like browsing, using messenger, accessing local or external storage in the network system, several OES applications use special purpose browser e.g. Safe Exam Browsers (SEB) or Respondus Lockdown Browser as one of solution. Unfortunately, this is compatible only for specific web based exam which offer a quiz mode. Besides, special purpose browser can be high cost and difficult to use for partial examinees. We need to design a cheaper and more practice way to handle this condition.

3. Cheating can be done by some examinees if they can set their own schedule and prepare their seating position when to take an exam. There is no referenced paper or application which is special solving this problem.

4. Until recently in several online exam application, password hash was established as sort of de facto standard to use MD5 hash algorithm for protecting passwords. It becomes so popular that various public hash databases appeared online like `http://www.md5decrypter.com`. We need to construct our online exam with another newest hash algorithm like SH3 algorithm.

5. We need a perfect distribution of random algorithm of questions because the shuffle methods used by some CMSs are not fully uniform distributions.

6. Collusion between examinee and administrator can be done if examiner could not access to the system even just looking and downloading the exam result.

In order to ensure fairness of online examination and solving some existing problems above, our challenge here is to make a secure online examination system application with several new features that are not owned by any other systems that we referenced. Table 3.1 shows a comparison of several online examination systems with our scheme.

Table 3.1: Features comparison of Online Examination Systems

Features	**Ours**	**SI**[1]	**LG**[14]	**CDS**[15]	**HB**[16]	**IRI**[17]
Browsing Guard	Yes	No	Yes	No	No	No
The Internet Messenger Guard	Yes	No	Yes	No	No	No
Time Limit	Yes	Yes	Yes	Yes	Yes	Yes
Local Data Accessing Prevention	Yes	No	Yes	No	No	No
Ext.Storage Accessing Prevention	Yes	No	Yes	No	No	No
Random Question	Yes	Yes	Yes	Yes	Yes	Yes
Random Scheduling	Yes	No	No	No	No	No
Random Seating	Yes	No	No	No	No	No
Bank Question	Yes	No	No	No	Yes	Yes
Question Analyzing	Yes	No	No	No	No	No
Collusion Prevention	Yes	No	No	No	No	Yes
E-Monitoring	No	Yes	No	Yes	No	No

Yes/No: Feature shown in the left column is/is not held.

3.3 Model and Implementation

In this section, we offer a secure web-based online examination system along with network design so that the system [18] is expected to prevent cheating and network security that often occurs, which is either done by the participants taking the exam or by persons outside the system trying to penetrate.

3.3.1 Web Security Design

We try to utilize a secure website, which follows the recommendation by [11] about online exam control procedure. We can see the detail of pages in Figure 3.2.

The examinee page consists of 3 sub pages which are Home, Take a Test and View Result. To access these pages, an examinee must have a registered user name and password. Home is the first page that can be accessed after successful login. The most important subpage for the examinee is Take a Test page. The examinee must be aware of some properties when this page has been accessed, which are:

1. The type of test that will appear on this page and which will be accessed by the examinee is the test that has been registered as the test program.

2. Be at the place (PC Client's identity) that has been determined by the system when registration process.

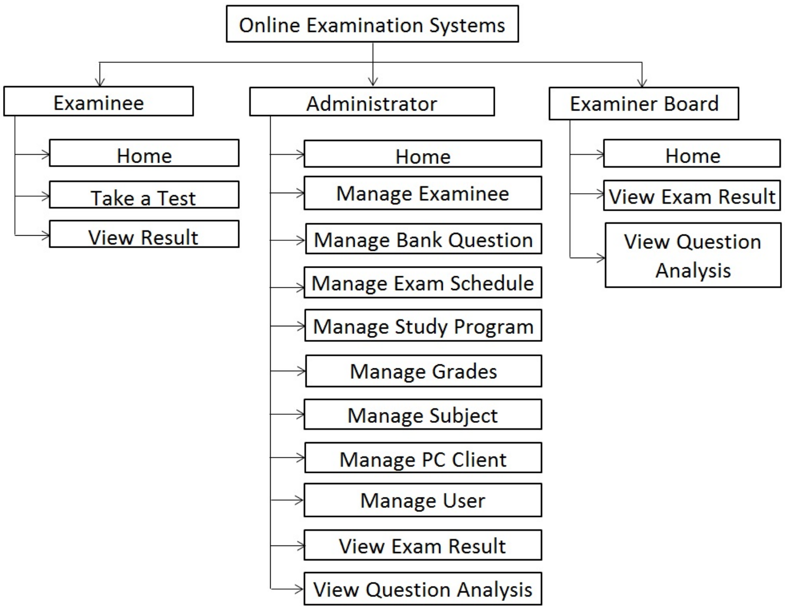


Figure 3.2: Online Examination System

3. Be at the time range that has been determined by the system when registration process.

4. After selecting the test subject, the timer will start and not be stopped until the time run out.

5. The questions will appear one by one on each page with the questions and candidates of answer appearing randomly for each examinee.

6. Each examinee can do the test only once. After that, the examinee will no longer be able to access the test questions.

7. Results are displayed for each examinee at the end of exam for automatically grading. Such a treatment rules out of manipulation of results. While for manually grading, the system will follow some procedure until the examinee gets his result (see: chapter 5.).

8. An honest examinee can see their result of the test but not that of other examinees.

9. Communication process of each processes will run on the particular secure protocol.

The second page is the administrator page. This page is the most important element of the online test system. In this page, all of the test terms are organized, such as the test schedule organization, examinee organization, inputting of test questions to the bank question, managing the study program, managing grades, managing subject, managing PC client which will be used by examinee to take an exam and manage of username and password. Besides, the administrator can view an exam result and question analysis when the examination has finished. It is recommended that there should be a limited number of people who can access the page for maximizing the security of data and system. The administrator has several terms too, which are:

1. Administrators can add, change and remove the question in a Bank Questions or Test Managing. In addition, we have feature for import questions from Excel file (*.xls or *.xlsx) with the particular format to the bank question.

2. Administrator can see the test result but they could not edit it.

3. The administrator registers an examinee but they could not see or edit the User name or password. User name and password are generated by the system at randomly and sent directly to the examinee's email when registration process.

4. The administrator manages an exam schedule but they could not see or edit the examinee's schedules and examinee's place to take the exam. Schedule and place are generated by the system at randomly and sent directly to the examinee's email when registration process.

5. The administrator could not deceive or make collusion with examinee because the result of an exam can be seen by the examiner board. Examiners board can be teachers association or school leaders.

The third page is examiner board page. It requires username and password authentication to access it, even though this page is only a viewing mode page. For those who can access this page, they can only view the test results and question analysis in several options like viewing the overall subject and all examinees for each grade, viewing daily results of each subject and viewing the question analysis. Questions Analysis page analyzes the difficulty of the question based on the answer of examinee and output of three categories of question, hard, moderate or easy. The Questions Analysis page can be used as a reference for teachers to know which materials are still not understood by the students.

3.3.2 General Network Security Design

Network design is also one of the main elements of the online examination system. We consider to easier and cheaper way to achieve goals both of security view and features view in OES. We have several points of interest in designing a network for security, which are:

1. All access to the web and online examination server database is blocked, except for access from registered proxy. It is hoped that this solution can become one of the guarantees for high security in overcoming illegal access from unauthorized users, as well as to prevent malpractice or illegal use.
2. All outgoing accesses of the client, by which the examinee is taking the examination, will be blocked except access to the online examination server. By blocking all accesses to the outside from client, it is hoped that cheating by looking for answers through the Internet or by using Messenger applications can be overcome.
3. The operating system of the client uses Windows OS which will restrict some actions during the examination, which are: prohibiting access to Windows explorer, prohibiting access to external ports such as USB port, CD/DVD drives, floppy drives, tape drives and others. This is done because cheating in examination has frequently occurred through obtaining answers from outside sources using drives on the client PC.

Figure 3.3 is a block diagram of the online examination system network that we suggest. In this diagram, we assume that the data is transmitted through the Internet using a secure system such as using https protocol or other protocols that can hide data from eavesdroppers.

3.3.3 System Implementation

a.Web Application

For security on the website programming side, we provide initial authentication facility which uses the user's login and password. Moreover, in the database, we hide several parts such as the password of each user that is used for login using hash function SHA-3 which is taken from `https://github.com/jedisct1/keccak-php`, as well as all exam questions stored in the database. We use mcrypt function of PHP with the most effective encryption algorithm in [42].

We achieve cheating prevention with several features by making random questions for each examinee algorithm, session system for one time login,

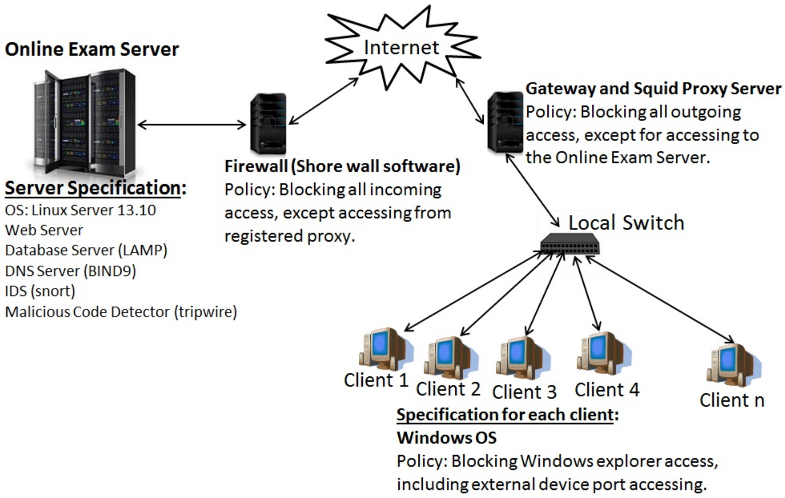

Online Exam Server
Internet
Gateway and Squid Proxy Server
Policy: Blocking all outgoing access, except for accessing to the Online Exam Server.
Firewall (Shore wall software)
Policy: Blocking all incoming access, except accessing from registered proxy.
Server Specification:
OS: Linux Server 13.10
Web Server
Database Server (LAMP)
DNS Server (BIND9)
IDS (snort)
Malicious Code Detector (tripwire)
Local Switch
Client 1
Client 2
Client 3
Client 4
Client n
Specification for each client:
Windows OS
Policy: Blocking Windows explorer access, including external device port accessing.

Figure 3.3: Network design of Online Examination Systems

question bank, random seat order, randomizing exam schedule and time limit algorithm for every question. This is based on a recommendation by [11]. We only will explain some new features which still not exist in the others system.

Random Question algorithm

Randomization algorithms of exam questions that are widely used by online examination systems today still use the shuffle function, which is owned by the programming language, such as shuffle() or rand() in PHP and random() in ASP. The weakness of this model is the randomness that is not completely uniform so that in the case of this online exam, there are questions that occur very frequently and there are those that rarely occur. One algorithm which is famous for its near-perfect uniform randomization is the Fisher-Yates Algorithm [19].

The Fisher-Yates shuffle (named after Ronald Fisher and Frank Yates) or also known as the Knuth shuffle (taken from the name of Donald Knuth), is an algorithm to generate a random permutation of a finite set, in other words, to shuffle that set. If implemented correctly, the results of this algorithm will not be biased, so that every permutation is equally likely. The basic method used to generate a random permutation of the numbers 1 through N is as follows:

1. Write number 1 to N.
2. Choose a random K between 1 and N that has not yet been scratched out.

3. Scratch out the K, and write that number in another location.
4. Repeat steps 2 and 3 until all numbers have been scratched out.
5. The order of number written in step 3 is the random permutation from the beginning numbers.

In the modern version currently used, the numbers chosen is not scratched, but its position is exchanged with the last digit of the numbers that have not been selected. Table 3.2 shows the flow of Fisher-Yates algorithm.

Table 3.2: Sample of Fisher-Yates algorithm Shuffle

Range	Roll	Scratch	Result
		12345678	
1-8	4	1238567	4
1-7	2	173856	4 2
1-6	5	17386	4 2 5
1-5	1	6738	4 2 5 1
1-4	3	678	4 2 5 1 3
1-3	8	67	4 2 5 1 8
1-2	6	7	4 2 5 1 8 7
Randomizing Result:			4 2 5 1 8 7 6

With a little modification from original Fisher-Yates algorithm, we construct our random question algorithm. Figure 3.4 illustrates the steps of our algorithms.

With this Fisher-Yates shuffle algorithm, we can assume that choosing the set of questions for each examinee is done by uniform random distribution. Let X = The number of questions in Bank question, Y = The numbers of questions at exam and Z = Total of examinees who takes an exam in the same place and time. Based on the permutation theory, we can compute several things below:

1. We can compute how many possibilities different set of questions if we choose Y questions from X total questions $(X > Y)$.
   $n = P(X,Y) = \frac{X!}{(X-Y)!}$
2. Then, we can compute how many probability of Z examinees for getting same set of question.
   $Pr[Q] = \frac{Z}{n} = \frac{Z}{P(X,Y)}$

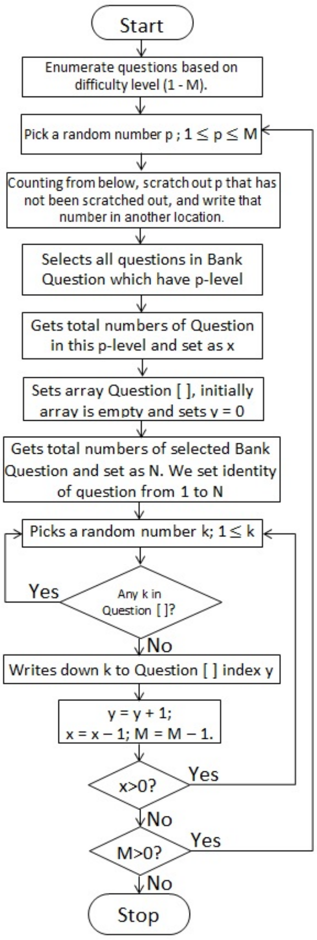

Start
Enumerate questions based on difficulty level (1 - M).
Pick a random number p ; 1 ≤ p ≤ M
Counting from below, scratch out p that has not been scratched out, and write that number in another location.
Selects all questions in Bank Question which have p-level
Gets total numbers of Question in this p-level and set as x
Sets array Question [ ], initially array is empty and sets y = 0
Gets total numbers of selected Bank Question and set as N. We set identity of question from 1 to N
Picks a random number k; 1 ≤ k
Yes
Any k in Question [ ]?
No
Writes down k to Question [ ] index y
y = y + 1;
x = x – 1; M = M – 1.
x>0?
Yes
No
M>0?
Yes
No
Stop

Figure 3.4: Flowchart of Random Question Algorithm

As an example, If there are 150 questions in the Bank Question and we will choose 100 questions for exam then we will have around $1,879x10^{198}$ possibilities of set questions. If we choose one of them for 10 examinees only, then they only have probability of getting same set question around $5.3233x10^{-198}$. It is very small probabilities, close to 0%.

Automatic scheduling and seating arrangement

In an automatic scheduling and seating arrangement system, we also use the randomization method. The system with database of identities of exam schedule and client PC selects an exam schedules, generates an exam both at random and provide them to examinee during registration, by checking whether the schedule and exam location have not been taken by examinees who have registered previously.

This process is performed by the system when examinee registered, along with the generation of a user name and password that will be used by the examinee to log in to OES. User name, password, exam schedules and exam's place are directly sent to the examinee's email address by the system without administrator intervention. In this case, an administrator is only in charge of inputting the personal data of the examinees into system. This idea is based on the suggestion from [20]. According to them, one of technique that can be used to reduce cheating on exams is automated sitting positions. Figure 3.5 is an illustration of the registration process that will generate a user name, password, automatic scheduling, and sitting position. If we assume that our automatic scheduling and seating arrangement algorithm is uniform random distribution, we can compute the probability of examinee who try to do cheating by get a seat position side by side. Let X = Total examinees who want to take an exam, Y = total available seats and Z = The numbers of dishonest examinees.

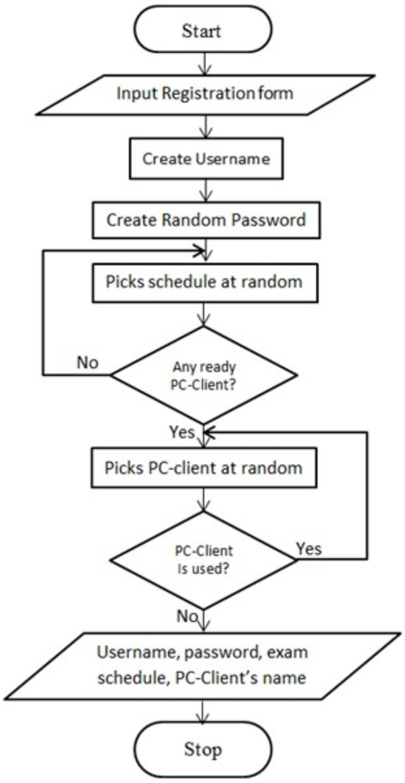


Figure 3.5: Flowchart of Registration Process

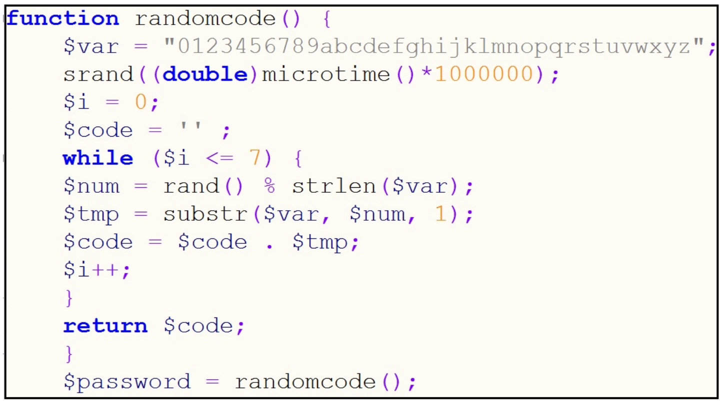
$Pr[R] = \frac{P((X-Z),(Y-Z))}{P(X,Y)}$

As an example, If there are 100 examinees and 10 available seats in one time exam. Then we can compute probability of 2 dishonest examinees will get side by side seats as follow :

$Pr[R] = \frac{P((100-2),(10-2))}{P(100,10)} = \frac{P((98),(9))}{P(100,10)} = \frac{98!}{100!} = 0.0001 = 0,01\%.$

To generate random passwords, we create sets using scrambler function which generates, for example, 8 digits derived from the numbers 0-9 and the letters a - z. Figure 3.6 is a function of the scrambler password.

Question analyzing algorithm



```
function randomcode() {
    $var = "0123456789abcdefghijklmnopqrstuvwxyz";
    srand((double)microtime()*1000000);
    $i = 0;
    $code = '' ;
    while ($i <= 7) {
    $num = rand() % strlen($var);
    $tmp = substr($var, $num, 1);
    $code = $code . $tmp;
    $i++;
    }
    return $code;
    }
    $password = randomcode();
```

Figure 3.6: Random Password Generating function

One advantage of our application compared to previous applications is a feature to analyze the question which exists on the Bank Question. After the exam is done, the examiner can see each question in the category of hard, medium or easy. This analysis is based on a comparison of the number of correct answers to the total numbers of answer which we call the difficulty index [21]. The index value is calculated using equation below.

$$p = \frac{R}{T}$$

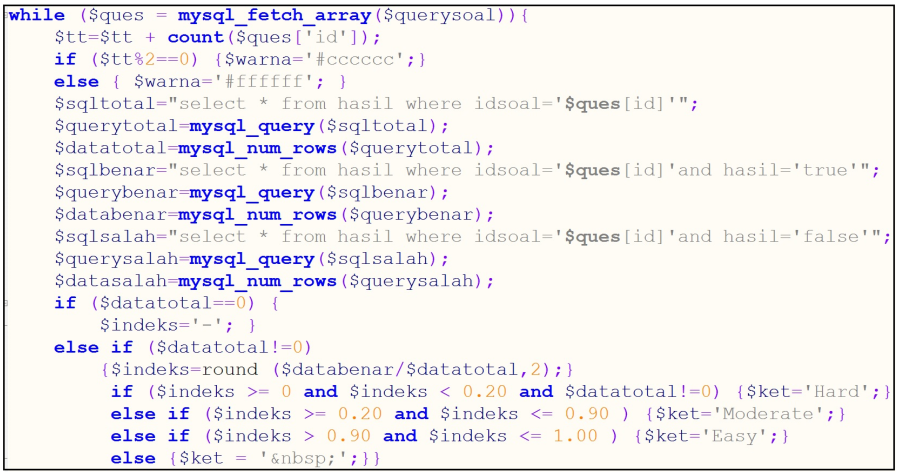
Where:

p = Difficulty Index

R = Number of correct answers to the exam and

T = Total numbers of the answer to the exam.

If the difficulty index is smaller than 0.2, then the question is considered hard. If the difficulty index is in the range 0.2 to 0.9, then the question is considered medium or moderate. While the difficulty index is bigger than 0.9, then the question is considered easy. The following figure 3.7 is our questions analysis function.



```
while ($ques = mysql_fetch_array($querysoal)){
    $tt=$tt + count($ques['id']);
    if ($tt%2==0) {$warna='#cccccc';}
    else { $warna='#ffffff'; }
    $sqltotal="select * from hasil where idsoal='$ques[id]'";
    $querytotal=mysql_query($sqltotal);
    $datatotal=mysql_num_rows($querytotal);
    $sqlbenar="select * from hasil where idsoal='$ques[id]'and hasil='true'";
    $querybenar=mysql_query($sqlbenar);
    $databenar=mysql_num_rows($querybenar);
    $sqlsalah="select * from hasil where idsoal='$ques[id]'and hasil='false'";
    $querysalah=mysql_query($sqlsalah);
    $datasalah=mysql_num_rows($querysalah);
    if ($datatotal==0) {
        $indeks='-'; }
    else if ($datatotal!=0)
        {$indeks=round ($databenar/$datatotal,2);}
         if ($indeks >= 0 and $indeks < 0.20 and $datatotal!=0) {$ket='Hard';}
         else if ($indeks >= 0.20 and $indeks <= 0.90 ) {$ket='Moderate';}
         else if ($indeks > 0.90 and $indeks <= 1.00 ) {$ket='Easy';}
         else {$ket = '&nbsp;';}}
```

Figure 3.7: Question Analysis Function

b.Network Configuration

Several OES use Safe Exam Browser or Respondus Lockdown Browser at the client side for security and cheating prevention. Creating special browser using ActiveX [14] is another way, but both of them need high costs and sometimes uncomfortable for all examinee. So, we remove these techniques to make easier and cheaper. Our solution is using proxy and firewall system, utilization of MMC (Microsoft Management console) and some additional features for security goal.

Server Security

In the server, all data and application for online examination are stored. We propose to use Linux server 14.04, using LAMP for web servers, using MySQL for database server, using BIND9 for DNS system and Shorewall [43] for Firewall system. The policy in Shorewall firewall server is to block all incoming access, except accessing from listed proxy servers. We have to set-up global rule to block all connection and a specified rule to accept access from specified network at Shorewall firewall server. Besides, we also optimize the configuration in iptables which is provided in the Linux kernel firewall to set up and maintain tables of IP packet filter rules in the Linux kernel. We have to set-up global and specified rule at Shorewall firewall server as figure 3.8. However to ensure security on the server side, it is not enough just with this configuration. Here, there are several addition features that we have used for the server side:

1. We have installed the Intrusion Detection System (IDS) which serves to check incoming and outgoing data packet activity in the network.

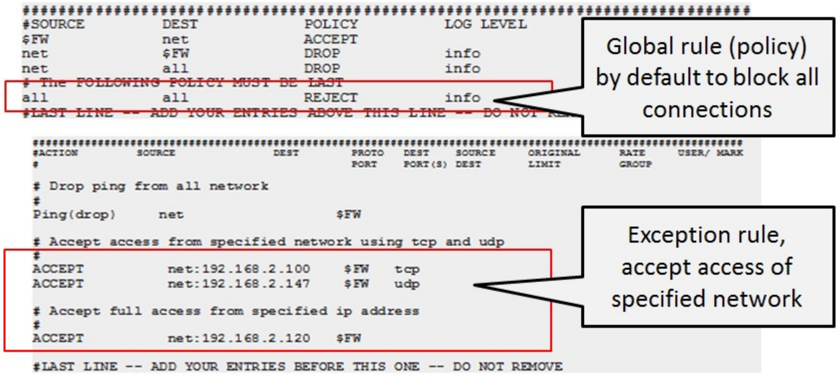
Besides, it can identify suspicious pattern that possibly happen in the network. We used "snort" software as IDS because it is open source GNU and it can also be modified as needed.

2. We have activated and configured the Malicious Code detector. Here, we used "tripwire" software which also open source and easy to configure.

3. We do not allow the user to use remote programs such as telnet. All remote programs are switched off after installation. To ensure security, the administrative processing must be done with locally accessing.

4. We added "Disallow: /administrator/" in robots.txt file or "Disallow: /[directory_name]/", where [directory_name] is a directories which does not want published in search engine. We need to protect these files as good as possible because it is always used by a hacker to know our website structure.

5. After completing development, we need to do restriction (chmod) for all files and directories starting from the root of our website. all access –rwx group should be disabled and accessing to "other" user should not be able to do "write" process unless directories like cache which is needed by the web server or directories where the file will be upload.



```
##################################################################################
#SOURCE          DEST             POLICY           LOG LEVEL
$FW              net              ACCEPT
net              $FW              DROP             info
net              all              DROP             info
# The FOLLOWING POLICY MUST BE LAST
all              all              REJECT           info
#LAST LINE -- ADD YOUR ENTRIES ABOVE THIS LINE -- DO NOT RE
```

```
###################################################################################################
#ACTION       SOURCE                 DEST        PROTO   DEST     SOURCE     ORIGINAL     RATE     USER/ MARK
#                                                PORT    PORT(S)  DEST       LIMIT        GROUP

# Drop ping from all network
#
Ping(drop)     net                   $FW

# Accept access from specified network using tcp and udp
#
ACCEPT              net:192.168.2.100    $FW   tcp
ACCEPT              net:192.168.2.147    $FW   udp

# Accept full access from specified ip address
#
ACCEPT              net:192.168.2.120   $FW

#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
```

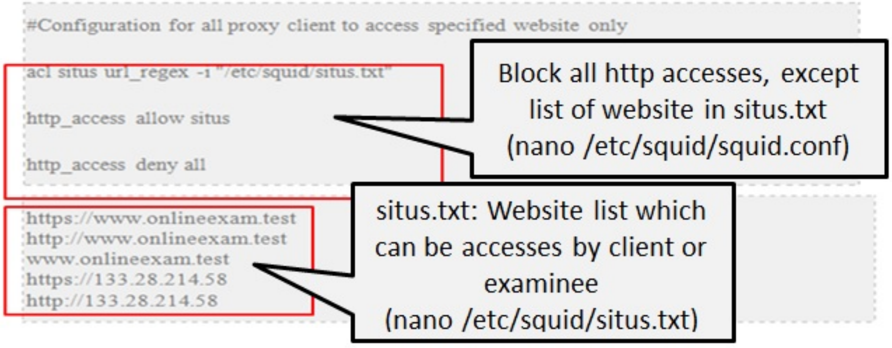
Global rule (policy) by default to block all connections

Exception rule, accept access of specified network

Figure 3.8: Global and Specified Rule of Firewall

Client Security

We adopt the squid proxy server because it has several advantages [44]. The policy in the proxy server is to block all http accesses, except access to the online examination web page. We have to set-up rule at proxy squid server to block all http accesses, except the websites listed in the proxy list. We can make a proxy list which can be accessed by client or examinee. We have to set-up rule at proxy squid server too as show in figure 3.9.



```
#Configuration for all proxy client to access specified website only

acl situs url_regex -i "/etc/squid/situs.txt"

http_access allow situs

http_access deny all
```

```
https://www.onlineexam.test
http://www.onlineexam.test
www.onlineexam.test
https://133.28.214.58
http://133.28.214.58
```

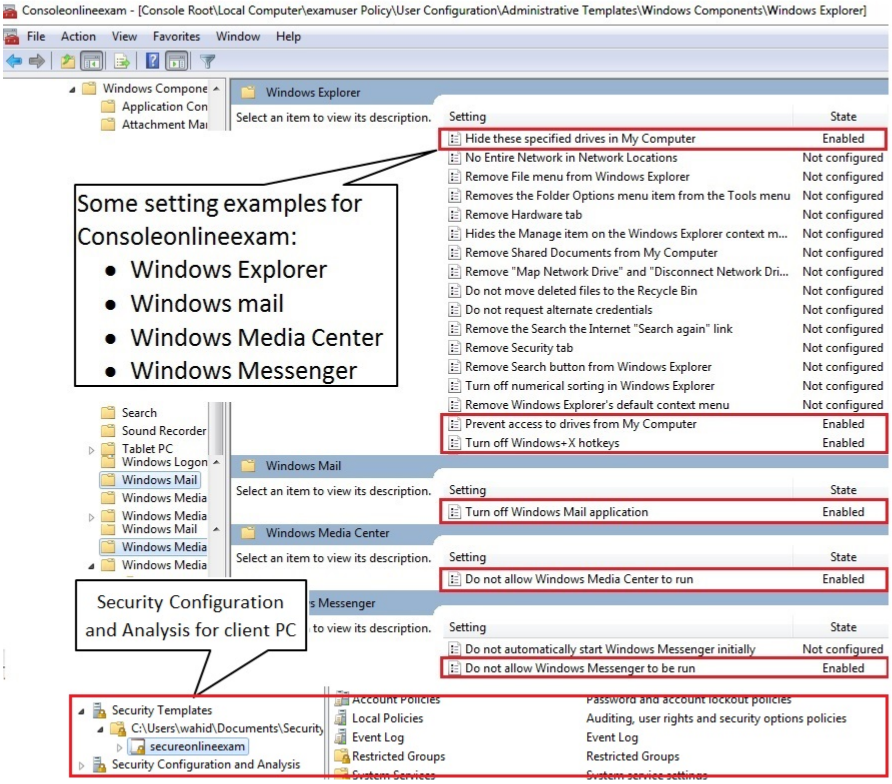
Block all http accesses, except list of website in situs.txt (nano /etc/squid/squid.conf)

situs.txt: Website list which can be accesses by client or examinee (nano /etc/squid/situs.txt)

Figure 3.9: Rule of proxy Squid Server

We assume that every client uses Windows operating system. In this system, we want to make a policy that each examinee which uses this system cannot access Windows explorer and external device port. Examine also can not run several application softwares. Besides, we propose to use Microsoft Management console (MMC) as a solution. It is a graphical user interface-based component in Windows that accommodates administrative tools called snap-ins [45].

As we have explained above, that we have prevented some cheating techniques from configuring some of the ”group policy” on the client computer to make a ”consoleonlineexam” using MMC. In addition, we also did some security configuration in order to ensure some level of security on the client side. In the ”consoleonlineexam”, we added the ”Security Template” and ”Security Configuration and Analysis” snap-in. This configurations aim to limit the examinee user group permissions to access system file and registry, so they cannot change the ”group policy” that has been set before. This configuration also assumes that the administrator does not want the user entered into the Power Users Group. The advantage of MMC is simpler for an administrator with enough once to configure and can be applied to all client computers in the network. Figure 3.10 is MMC configuration in the ”consoleonlineexam”. Besides, it can configure or analyze Windows operating system security. Its operation is based on the contents of a security template that was created using the Security Templates snap-in.



Some setting examples for Consoleonlineexam:

- Windows Explorer
- Windows mail
- Windows Media Center
- Windows Messenger

Security Configuration and Analysis for client PC

Windows Explorer

Select an item to view its description.

Setting	State
Hide these specified drives in My Computer	Enabled
No Entire Network in Network Locations	Not configured
Remove File menu from Windows Explorer	Not configured
Removes the Folder Options menu item from the Tools menu	Not configured
Remove Hardware tab	Not configured
Hides the Manage item on the Windows Explorer context m...	Not configured
Remove Shared Documents from My Computer	Not configured
Remove "Map Network Drive" and "Disconnect Network Dri...	Not configured
Do not move deleted files to the Recycle Bin	Not configured
Do not request alternate credentials	Not configured
Remove the Search the Internet "Search again" link	Not configured
Remove Security tab	Not configured
Remove Search button from Windows Explorer	Not configured
Turn off numerical sorting in Windows Explorer	Not configured
Remove Windows Explorer's default context menu	Not configured
Prevent access to drives from My Computer	Enabled
Turn off Windows+X hotkeys	Enabled

Windows Mail

Select an item to view its description.

Setting	State
Turn off Windows Mail application	Enabled

Windows Media Center

Select an item to view its description.

Setting	State
Do not allow Windows Media Center to run	Enabled

s Messenger

to view its description.

Setting	State
Do not automatically start Windows Messenger initially	Not configured
Do not allow Windows Messenger to be run	Enabled

Account Policies	Password and account lockout policies
Local Policies	Auditing, user rights and security options policies
Event Log	Event Log
Restricted Groups	Restricted Groups
System Services	System service settings

Figure 3.10: Microsoft Management Console Configuration

3.4 Security Considerations

As we have mentioned in section 3.1, the problems faced by the online examination system of several previous researches are security issues and the problem of cheating. The following is an explanation of how to solve some problems by the techniques offered in section 3.3.

3.4.1 Security and Reliability

a. Server Side

For security and reliability, we have taken the following features to our server side system:

1. Database secrecy; We carried out encryption on important data stored in the database. This is to prevent data theft when someone has successfully accessed the database illegally. Password data is hidden by

hash function SHA-3, and the examination questions are encrypted by mcrypt function at PHP with algorithm XTEA.

2. Authority system; Systems authorize different authorities for each user, and the users use the system within their own authorities. We try to categorize our system into three groups; administrator, examinee and examiner board. A session ID is embedded to each authorizing users for preventing illegal access. The system can protect the questions and test sheets in the database from deliberate steal or alteration. With the perfect authentication and authorization, users are limited to authorized functions, so that the security of data in the system can be guaranteed.

3. Data access control; Firewall will control for accessing data to the server. It blocks to access the web server from outside unless accessing from the registered proxy. Firewall guarantees that server only can be accessed by true examinees. Protection with "Disallow: /administrator/" command in the robots.txt file will hide our essential directories from a search engine in order to hide from bad hackers. In addition, restriction configuration "chmod" for all files and directories after development guarantees that no bodies can access, modify or delete it.

We perform a simulations attacks and testing aimed to determine the ability of all parts of the system. Attacks simulation with an active firewall (Shorewall) is shown in table 3.3. The result shows that the attacker cannot know the number of open ports on the target (Online Exam server). It can be concluded that Shorewall can hide the open port from viewed by computer attacker. So the chances of attacks against the server online examination beginning with port scanning can be overcome.

b. Client Side

In the client side, we have applied the client security control. Here, MMC configuration can controls securities in the computer client. We configure to restrict user for access system files or registry to modify or delete a group policy which has been set by the administrator.

3.4.2 Cheating Prevention

a. Server side

We consider cheating prevention on the application program. The following programs are used for the purpose.

1. Login system; This is the security standard for the online examination system. To prevent exam page access by other examinees, or accessing

Table 3.3: Scanning port simulation attack

Scanning port	Active Firewall	Not active Firewall
`nmap -T4 -A -v`	Not identified	Port number : 80,135,139,443,445,902,912,1025,1026,1027,1028,1036,2869,3306 and 10243
`nmap -T4 -sS -v`	Not identified	80,135,139,443,445,902,912,1025,1026,1027,1028,1035,1036,2869,3306 and 10243

to the administrator page by unauthorized individuals, a login system using username and password is required. It was hoped that the passwords are only known by authorized people and that it is periodically changed. We develop every page by embedding a session ID which is different for each examinee after they login to the system.

2. One time exam system; This algorithm only will limit the examinee to be able to take the exam only once. If they have already completed the examination, they will not be able to access exam question again. In this way, the possibility of exam questions being leaked is minimized.

3. Fisher-Yates Random question; The purpose of randomizing questions is to randomize the exam question so that each examinee's examination questions will have a different order. By the Fisher-Yates shuffle algorithm, we can make a fully uniform random question system and we give guarantee that all examinee will get question with different order. There is no examinee will get same order questions. This will minimize the cheating probability by exchanging answers.

4. Accessing period; Access to exam questions can only be done during a certain period. Outside the time period, the examinee can no longer access the questions.

5. Time limit; Each question will have a time limit to complete, so that it becomes less possible to exchange answers or discuss between examinees.

6. Bank Question; If the numbers of questions which is provided in the database, much larger than the numbers of question which will be

tested, then the opportunities for questions similarities between examinee and the others examinee will be smaller. So Bank Question is one way to prevent the cheating between examinees.

7. Random exam scheduling and seating arrangement; In order to decrease opportunities for test collusion between examinee, the system must be designed to perform scrambling on exam seats and exam schedules. Thus, examinees cannot determine when they will take an exam and where they will sit in order to be close to friends. The study concerned about cheating at Midlands State University (MSU) [20] suggests using automated sitting positions. In their questionnaire responses, respondents indicated that it can be used to reduce cheating. In general, this feature aims at reducing the human work by arranging the exam schedules and seats automatically, and to obtain an optimized seating arrangement where preventing two colluded students already have agreed to sit next to each other.

b. Client side

There are several kinds of restriction in client side to prevent cheating. In the proxy system, the examinee will be blocked to access another website besides the online examination system website. Its purpose is to prevent the examinee to find answers using the search engine or other facilities of the Internet. In previous research, we have to use a specific browser like Respondus Lockdown Browser or safe exam browser to prevent access to another website.

Table 3.4 and table 3.5 show the results of simulation testing by accessing various websites either using http, https or ftp, by writing the port number used or without the port number. The result is that when a proxy server is enabled then only the IP of the online examination system that can be accessed while the DNS than that cannot be accessed. Conversely, if a proxy server is made inactive, then all of the tested DNS addresses can be accessed unless address of the online examination. It is concluded that the function of the proxy server which is installed on the client can running well.

Besides that, we adopted to use Microsoft Windows operating system for each client who wants to take an examination. The MicrosoftWindows there is a facility which can help us to prevent the cheating by examinee. We only need to set up and configure MMC.exe file to activate this function and then we can prevent several common cheating technique like examinee cannot access local and external storage using several drives like USB port, CD/DVD drives, floppy drives.

The next experiment is testing the MMC settings to restrict access to resources of examinees or computer client whether it is devices or application

Table 3.4: Active proxy Server testing

URL or IP Address	Result
http://133.28.214.58 (Online exam)	Success
http://133.28.214.58:80 (Online exam)	Success
https://133.28.214.58/ (Online exam)	Success
https://133.28.214.58:443/(Online exam)	Success
http://www.google.com	Not Success
http://www.yahoo.com	Not Success
http://ppiishikawa.org:2095/	Not Success
ftp://ftp.ppiishikawa.org	Not Success
ftp://ftp.ppiishikawa.org:21	Not Success

Table 3.5: Not active proxy Server testing

URL or IP Address	Result
http://133.28.214.58 (Online exam)	Not Success
http://133.28.214.58:80 (Online exam)	Not Success
https://133.28.214.58/ (Online exam)	Not Success
https://133.28.214.58:443/(Online exam)	Not Success
http://www.google.com	Success
http://www.yahoo.com	Success
http://ppiishikawa.org:2095/	Success
ftp://ftp.ppiishikawa.org	Success
ftp://ftp.ppiishikawa.org:21	Success

programs, and the results are presented in table 3.6. It shows that we can prohibit access to external devices by utilizing the functionality of Microsoft Management console (MMC). MMC is already available on Microsoft Windows which used by the client. Using some default application programs from Microsoft Windows can be made disable like Windows Messenger, Windows mail, Windows media player and so on.

3.4.3 Additional features

In order to make easier for participant, especially for administrator and examiner board, we provided several additional features:

Table 3.6: MMC testing simulation

Access device or program	Result
`Accessing Windows Explorer`	Blocked
`Using USB flashdisk drives`	Blocked
`Using CD/DVD drives`	Blocked
`Using Floppy drives`	Blocked
`Using Windows+X hotkeys`	Blocked
`Using Windows Messenger`	Blocked
`Using Windows mail`	Blocked
`Using Windows media player`	Blocked

1. Questions analyzing; Question analyzes feature can help administrator or examiner board to analyze the result of the exam. From this, the examiner will know where the difficult, moderate or easy question based on test result. It can be a reference for make improvisation to teach or make a question in the future.

2. Auto generated downloadable file system; Auto generated downloadable file feature will help administrator to make a report for examination result or question analysis result. It is very helpful to provide a physical report which is sometimes needed for the formally report.

3. Bank Question file import system; This feature allows the administrator to enter the question data into the database from files with *.xls or *.xlxs extensions using a predetermined format without having to input one by one. It is also very helpful for maintaining the secrecy of questions from others person. With this feature, Administrator only needs less time to transfer questions for the database.

Unfortunately, Website application and network design cannot meet some security requirements because of malicious behaviors of bribed, corrupted or unfair examiners, dishonest or untrusted exam authority and several inside and outside attacks, and we construct a particular online examination protocol to prevent them.

Chapter 4

Certificateless Signcryption Scheme

Security protocol is one of the most important mechanisms in providing security of public networks because crucial information is hidden by this mechanism. In the design of security protocols, security issues and efficiency are a major concern. Concerning the security issues, we should consider at least the following 4 major security issues of networks, namely:

1. Confidentiality: Confidentiality ensures that the data or information cannot be accessed by unauthorized users.

2. Integrity: Integrity ensures that the data or information cannot be modified during delivery.

3. Authenticaton: Authentication has to guarantee both of the authenticity of user and the authenticity of data. User authentication ensures that the user who can access the system is a true user, while data authentication ensures that the received data actually comes from the true sender.

4. Non-repudiation: Non-repudiation ensures that user cannot deny the data or information which she or he has sent.

Message encryption schemes and digital signature schemes are cryptographic tools used for confidentiality, integrity, authentication and non repudiation [4]. Confidentiality can be achieved by the encryption schemes. Integrity, authentication, and non-repudiation can be achieved by the digital signature schemes. In this chapter, we study a special type of cryptographic protocol called certificateless signcryption scheme, which satisfies all of the 4 properties described above.

4.1 Background

Public-key cryptosystems are one of cryptographic protocols widely used nowadays. In the systems, each user chooses their own private key to calculate its corresponding public key. In the use of the public key, one should be able to know whose public key one is using. A certificate issued by a certification authority shows this connection between the public key and users identity. Thus, the systems require Public Key Infrastructure (PKI) whose bandwidth consumption and maintenance cost are usually high.

In order to reduce the burden caused by the PKI, Identity-Based Cryptography (IBC) was invented. This concept was first discovered by Adi Shamir [22] as well as Tatsuaki Okamoto [23] and its secure and efficient technique was recently discovered by Boneh and Franklin in 2001 [24]. In the IBC, the identity of a user such as name, ID number, email and telephone number serves as the public key, and there is no longer any doubt about the authenticity of the public key. Therefore, PKI can be eliminated. In addition, this technique easily allows one to set the validity period of the keys without requiring an additional key revocation mechanism. Despite these advantages, the private key of the IBC is generated from users identity by the Private Key Generator (PKG) and a key escrow problem inherently exists in the IBC.

Certificateless cryptosystem (CLC) [25] which is a variant of the IBC is intended to prevent the key escrow problem. In the ordinary IBC, keys are generated by key generation center (KGC) which is given a complete power and is fully trusted. In contrast, the CLC considers a compromised KGC. To prevent a complete breakdown of the system under the compromised KGC, the key generation process is split between the KGC and each user. First, the user generates a random value which is never revealed to anyone, including the KGC, as in the public-key cryptosystem. Then the KGC generates a private key based on the identity of the user, where the private key is now a partial private key of the system and sent to the user. Afterwards, all cryptographic operations by the user are performed by using a complete private key which involves both the partial private key and the user's random secret value. Therefore, the best features of the IBC and the public-key cryptosystem are combined.

On the other hand, there are several approaches such as signcryption and Elliptic Curve Cryptography which focus more on the efficiency. Signcryption introduced by Yuliang Zheng in 1997 [26] is a technique in which the functions of digital signature and encryption are achieved in just one logical step. It is effective in reducing computational cost and communication overhead compared to the signature-then-encryption technique. So far, there have been many studies on the signcryption. Elliptic Curve Cryptography (ECC) is based on the algebraic structure of elliptic curve over a finite field [27]. ECC has become a very important part in cryptography because of its high performance by a shorter key with the same level of security as other public key techniques. The Elliptic Curve-Discrete Logarithm Problem (EC-DLP) and Elliptic Curve-Computational Diffie Hellman Problem (EC-CDHP) can be defined in the ECC, and one can construct security protocols based on these problems [28, 29].

The efficiency of the CLC can be improved by applying these approaches. Actually, in 2008 Barbosa and Farshim [30] proposed a Certificateless Signcryption (CLSC) scheme which combines the CLC and the signcryption scheme. So far, several CLSC schemes have been proposed. However, most of the schemes are based on bilinear pairings. The time needed for running bilinear pairings is about 10 times slower than that needed for running the finite field exponentiation algorithm [31]. In order to overcome such an efficiency problem, more efficient CLSCs based on the finite field exponentiation have been offered by [31, 32] without using bilinear pairings. Even so, the computation of finite field exponentiations as well as bilinear pairings need large integer values for keeping the complexity of problems related to them. Under limited resource environments such as low memory and power consumption with constrained bandwidth, we need to find a more efficient construction.

In this chapter, we further apply the ECC to the framework of the CLSC and construct a more efficient CLSC. To this end, we do not modify existing CLSCs but construct a new certificateless signcryption scheme based on elliptic curve cryptography from scratch. In the design of our construction, we pay attention to certificateles hybrid signcryption schemes [28, 29] explained in the next section. The proposed scheme provides confidentiality, authentication, integrity, non-repudiation as well as unforgeability and forward secrecy. Since it is one of CLSC schemes, it solves the certificate management problem and the key escrow problem. By the evaluation of our CLSC scheme via the implementation and other analysis, we shows that our CLSC scheme has a better efficiency than existing schemes in terms of the ciphertext size and the execution time of key generation, signcryption, and unsigncryption phases.

4.2 Related Work

In 1997, Zheng [26] offered a primitive cryptographic technique that carries out both digital signature and message encryption functions simultaneously which he called signcryption. The cost of signcryption is much smaller than the signature-then-encryption model. There are several signcryption schemes [33, 34, 35, 36] that have been proposed since 1997. One of them is a signcryption scheme proposed by Zheng and Imai [36] that utilizes the hardness of EC-DLP. They proved that this signcryption scheme has an efficiency of approximately 58% of the computational cost and 40% of communication cost the signature-then-encryption scheme based on an elliptic curve.

In the signcryption scheme, the user's public key is a random element of some group. Therefore, this scheme does not provide user authentication itself because the random group element cannot define the identity of the user. This problem can be solved by the use of certificates, where there is a CA that provides a setting in which the public key is bound to the identity of each user. This system is known as the PKI. However, PKI has difficulties in the manufacture, storage, and distribution of its digital certificates.

To overcome these issues, Shamir [22] introduced the concept of Identity-based cryptography. The main idea is that the identity information such as name, e-mail, telephone number, or identity number of each user is used as its public key and not derived from certificates issued by the CA. In Identity-Based Cryptography, users can perform secure communications without the need to distribute public key certificates, without the need to store a public key directory and without the participation of online Public Key Generator (PKG). In addition to that already offered by Chen and Malone-Lee in 2005 [34], there are already some identity-based signcryption schemes that have been offered [34, 37, 38]. Unfortunately, their work still has the disadvantages that key escrow problem such that PKG holding all secret keys of the system has to be fully trusted.

Certificateless cryptography [25] is proposed to solve this key escrow problem. As a variant of Identity-Based Cryptography, certificateless cryptography uses a users identity as a public key and the KGC generates a partial private key of the user from th identity. Another private key is created by the user. KGC is not fully trusted because it does not know the whole of users private key.

The certificateless schemes that uses the elliptic curve approach has been proposed in papers [28, 29]. These papers propose a certificateless hybrid signcryption scheme, called CLSC-TKEM (CLSC-tag Key Encapsulation Mechanism), to encapsulate keys that are shared by the sender and the recipient. The concept is that the sender will create a session key using a random value and the recipient's public key. The sender then sends out a public value that has a relation with the random value along with the digital signature to the recipient. The receiver then calculates the session key by using the public key along with the receivers private key. We adopt this concept with slight modifications and also incorporate the concept of signcryption to consruct a

Certificateless Signcryption protocol scheme based on Elliptic Curve.

4.3 Certificateless Signcryption

In this section, we offer the use of Certificateless Signcryption (CLSC) based on elliptic curve cryptography without pairing function. A scheme based on the concept of Barbosa-Farshim scheme [30] which can accept ID input and message of any length as well as use a secure one-time symmetric key encryption scheme and collision resistance hash function.

4.3.1 Formal Model CLSC

According Barbosa-Farshim scheme, certificateless signcryption is separated to six-tuple of probabilistic polynomial-time algorithms. Four of these algorithms are corresponding to key management operations, while two algorithms are identical to signcryption and unsigncryption algorithms. The detail algoritms are the following steps:

1. Setup(1^{κ}). This is a global set-up algorithm, which takes as input the security parameter 1^{κ} and returns the KGCs secret key *msk* and global parameters pars including a master public key P_{pub} and descriptions of message space $M(pars)$, cipher text space $C(pars)$ and randomness space $Ram(pars)$. This algorithm is executed by the KGC, which publishes *pars*.

2. $Extract - PPK(ID, msk, pars)$. An algorithm which takes as input $msk, pars$ and an identifier string $ID \in \{0, 1\}^*$ representing a users identity, and returns a partial private key d. This algorithm is run by the KGC, after verifying the users identity.

3. $Gen - SV(ID, pars)$ An algorithm which takes an identity and the public parameters and outputs a secret value x and a public key P. This algorithm is run by a user to obtain a public key and a secret value which can be used to construct a full private key. The public key is published without certification.

4. $Set - SK(d, x, pars)$. A deterministic algorithm which takes as input a partial secret key d and a secret value x and returns the full private key SK. Again, this algorithm is run by a user to construct the full private key.

   The signcryption and Unsigncryption algorithms are as follows:

5. $SC(m, SK_A, ID_A, PK_A, ID_B, PK_B, pars, r)$. This is the signcryption algorithm. On input of a message $m \in M(pars)$, senders full private key SK_A, identity ID_A and public key PK_A, the receivers identity ID_B and public key PK_B, the global parameters *pars* and possibly some randomness $r \in Ram(pars)$, this algorithm outputs a ciphertext $c \in C(pars)$ or an error symbol $\perp$.

6. $USC(c, SK_B, ID_B, PK_B, ID_A, PK_A, pars)$. The deterministic unsigncryption algorithm. On input of a ciphertext c, receivers full private key SK_B, identity ID_B and public key PK_B, the senders identity ID_A and public key PK_A and the global parameters *pars*, this algorithm outputs a plaintext m or a failure symbol $\perp$.

4.3.2 Proposed CLSC based on Elliptic Curve

This scheme modifies the Elliptic Curve Cryptography based Certificaless Hybrid Encapsulation Key scheme without Pairing [28] and the eCLSC-TKEM [29] to obtain all the advantages of both techniques. The scheme consists of three parts, namely Key Generator Center (KGC), Sender and Receiver. KGCs function is to calculate the partial private key and public key pairs for all users when they first join the system. This process is performed only once at the beginning and can be done offline. For the process, we divided this scheme into seven phases, setup-parameter which is run by KGC, Set Secret value which is run by each user, Partial private key extract which is run by KGC, Set Private Key, Set Public Key which are run by each user, Signcrypt which is run by sender and Unsigncrypt phase which is run by receiver. In the initial phase, the system will select and publish all elliptic curve security parameters for all users that exist in the system. Figure 4.1 shows our proposed protocol.

The following are details of the process of the system:

1. Set-Up Parameter: It is run by the KGC. KGC selects and publishes system security parameters as follows:

    - F_q = Finite field of large prime number q
    - (a, b) = elliptic curve value $< q$, satisfy to $4a^3 + 27b^2 \neq 0$ and $q \neq 0$
    - E/F_q = elliptic curve over finite field, satisfy to $q : y^2 = x^3 + ax + b \mod q$
    - G_q = a generator of EC

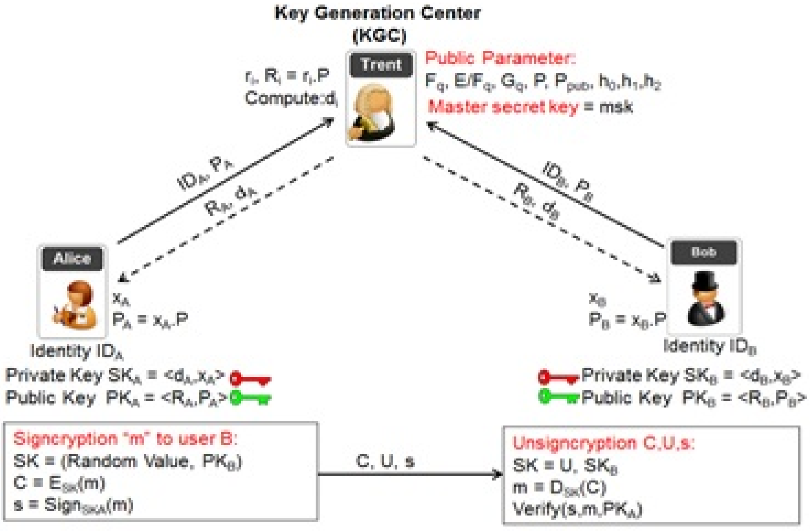


Figure 4.1: Certificateless Signcryption Protocol

- O = infinity point of EC, n is the order of F satisfy to $n.G = O$
- Hash function $h0 = \{0,1\}^* \times G_q^2 \rightarrow Z_q^*$
- Hash function $h1 = \{0,1\}^{*2} \times G_q^2 \rightarrow Z_q^*$
- Hash function $h2 = G_q^2 \times \{0,1\}^* \times G_q^2 \rightarrow Z_q^*$
- After that, PKG chooses integer $msk \in Z_q^*$ as the master secret key and calculate $P_{pub} = msk.G_q$ as master public key.
- PKG then publishes the public parameters $(F_q, E/F_q, G_q, h0, h1, h2, P_{pub})$ but keeps secret the msk.

2. Set secret value: It is run by each user. User i with ID_i performs the following steps:

   - Chooses randomly $x_i \in Z_q^*$
   - Computes public key $P_i = x_i.G_q$

3. Partial private key extract: It is run by KGC. Here, KGC produce the partial private key of every user based on their identity. The KGC processes the user i with ID_i in the following step:

   - Chooses randomly $r_i \in Z_q^*$ and computes $R_i = r_i.G_q$
   - Computes public key $d_i = r_i + msk.h0(ID_i, R_i, P_i).mod\ q$

- Sends to user $\langle R_i, d_i \rangle$ in a secure channel
- User Validate $d_i.G_q = R_i + h0(ID_i, R_i, P_i).P_{pub}$

4. Set Private Key: It is run by each user. User i with identity ID_i performs to set a private key pair $Sk_i = \langle d_i, x_i \rangle$

5. Set Public Key: It is run by each user. User i with identity ID_i performs to set a public key pair $Pk_i = \langle R_i, P_i \rangle$

6. Signcryption Alice is the sender. She wants to send message m to Bob as the receiver with identity ID_B, and a pair public key (R_B, P_B). Alice chooses $l_A \in Z_q^*$, then Alice computes :

   - $U = l_A.G_q$
   - $Y_B = R_B + h0(ID_B, R_B, P_B).P_{pub}$
   - $SK = h2(l_A.(Y_B + P_B), U, ID_B, R_B, P_B)$
   - $C = E_{SK}(m, ID_A)$
   - $s = (d_A + l_A.h1(m, ID_A) + x_A.h1(m, ID_A)).mod\ q$
   - Alice sends to Bob chipertext $= (C, U, s)$

7. Unsigncryption Bob is the receiver. He receives $= (C', U', s')$ from Alice. Bob computes:

   - $SK = h2((d_B + x_B).U, U, ID_B, R_B, P_B)$
   - $(m, ID_A) = D_{SK}(C')$
   - $Y_A = R_A + h0(ID_A, R_A, P_A).P_{pub}$
   - Verify: Accept if $s.P = Y_A + U.h1(m, ID_A) + P_A.h1(m, ID_A)$ is hold

4.4 Implementation in Javascript

In this section, we have implemented the simulation of our proposed scheme using JavaScript to test its truth. All of the security parameters use large numbers to protect the system from various types of attacks.

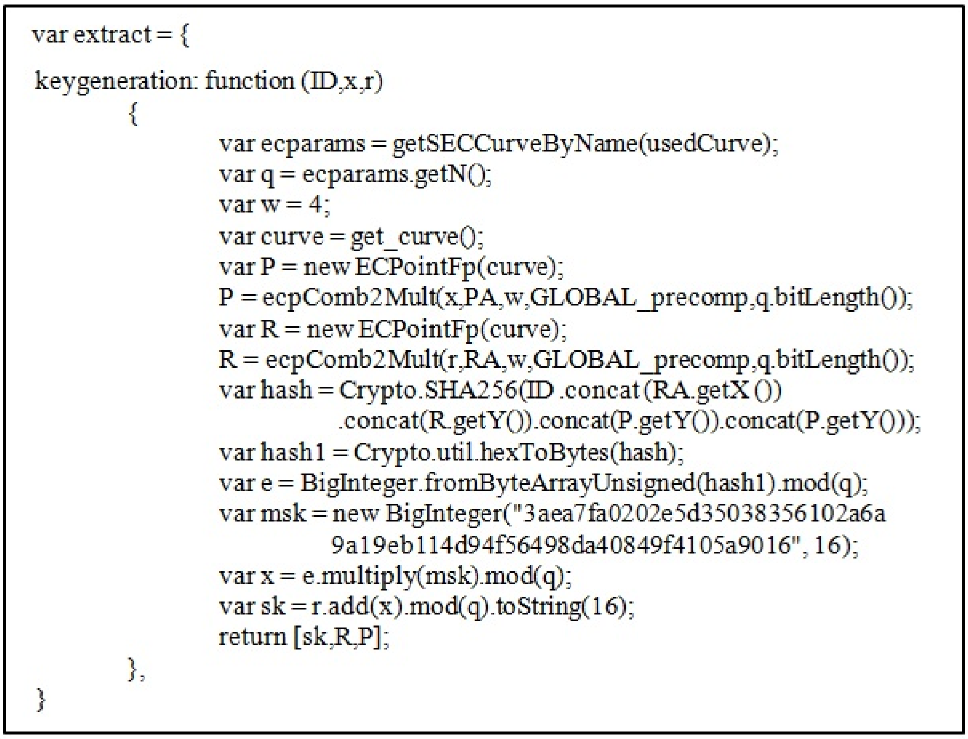
Here, we applied three types of elliptic curve function that are 512 bits brainpoolP512t1, 512 bits brainpoolP512r1 and 256 bits brainpoolP256. This value we are taken from
`https://github.com/spruegel/Fast-ECDSA-in-JavaScript/blob/master/`

Table 4.1: Elliptic curve values of our implementation

EC types	Var	Value (Hexadecimal)
`brainpoolP256r1()`	p	a9fb57dba1eea9bc3e660a909d838d726e3bf623d52620282013481d1f6e5377
	a	7d5a0975fc2c3057eef67530417affe7fb8055c126dc5c6ce94a4b44f330b5d9
	b	26dc5c6ce94a4b44f330b5d9bbd77cbf958416295cf7e1ce6bccdc18ff8c07b6
	n	a9fb57dba1eea9bc3e660a909d838d718c397aa3b561a6f7901e0e82974856a7
	P	8bd2aeb9cb7e57cb2c4b482ffc81b7afb9de27e1e3bd23c23a4453bd9ace3262, 547ef835c3dac4fd97f8461a14611dc9c27745132ded8e545c1d54c72f046997
	msk	3aea7fa0202e5d35038356102a6a9a19eb114d94f56498da40849f4105a9016
	P_{pub}	9a78b67f611ad4eb7d19d460cd4cc0e180d358c85a8212391bb266b1ab0ddb38, 67fb43aff1bf1a893520df1ef63145a9507856acce15061d6325c85c3ab0c7c3
`brainpoolP512r1()`	p	add9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca703308717d 4d9b009bc66842aecda12ae6a380e62881ff2f2d82c68528aa6056583a48f3
	a	7830a3318b603b89e2327145ac234cc594cbdd8d3df91610a83441caea9863bc 2ded5d5aa8253aa10a2ef1c98b9ac8b57f1117a72bf2c7b9e7c1ac4d77fc94ca
	b	3df91610a83441caea9863bc2ded5d5aa8253aa10a2ef1c98b9ac8b57f1117a72 bf2c7b9e7c1ac4d77fc94cadc083e67984050b75ebae5dd2809bd638016f723
	n	aadd9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca703308705 53e5c414ca92619418661197fac10471db1d381085ddaddb58796829ca90069
	P	81aee4bdd82ed9645a21322e9c4c6a9385ed9f70b5d916c1b43b62eef4d0098ef f3b1f78e2d0d48d50d1687b93b97d5f7c6d5047406a5e688b352209bcb9f822, 7dde385d566332ecc0eabfa9cf7822fdf209f70024a57b1aa000c55b881f8111b2 dcde494a5f485e5bca4bd88a2763aed1ca2b2fa8f0540678cd1e0f3ad80892
	msk	233276ac0ac1417aad31bab918f8b4676f0eca401343e2adfb154126a7df47ea90 7083357104431c1d0b12a61a85ac9561955783aa2ad71247c5a8ce3b3005e0
	P_{pub}	46fa83aee0bdb5e1197ef8b571a05b0f47ef44efd3ec6a8b739b0a72fd13c945a7 8d82a8ff1fc00949aecf47db237efa63f3edcc2e130d74ae2c1d80f31c577cl, 72a279fc2d13b3b54ff3434ad0ad85a8ff830fca4bc24b3b7260cc2f659711825e 61b3598717cc33fec53e0b970af07aab2d2623b5de98a7a89df234520be0d5
`brainpoolP512t1()`	p	aadd9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca703308717 d4d9b009bc66842aecda12ae6a380e62881ff2f2d82c68528aa6056583a48f3
	a	aadd9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca703308717 d4d9b009bc66842aecda12ae6a380e62881ff2f2d82c68528aa6056583a48f0
	b	7cbbbcf9441cfab76e1890e46884eae321f70c0bcb4981527897504bec3e36a62 bcdfa2304976540f6450085f2dae145c22553b465763689180ea2571867423e
	n	aadd9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca703308705 53e5c414ca92619418661197fac10471db1d381085ddaddb58796829ca90069
	P	640ece5c12788717b9c1ba06cbc2a6feba85842458c56dde9db1758d39c0313d 82ba51735cdb3ea499aa77a7d6943a64f7a3f25fe26f06b51baa2696fa9035da, 5b534bd595f5af0fa2c892376c84ace1bb4e3019b71634c01131159cae03cee9d 9932184beef216bd71df2dadf86a627306ecff96dbb8bace198b61e00f8b332
	msk	8e20b5a49ee25eec72bf9371da99b07156c7f11128b8607807ff5f347d6f80176c 576fcea2bb29540920fc8a2a71c925910d98def772276e706fa4b5c4d4fe32
	P_{pub}	8a27e89f37d19598bf4a4069b5cfdf25d54e5c3e931a1f26dcc862ed110f91a557 8ffcc04417b3af4fe6909c26d7abfba6291d1415533ddc2ebaea41ff6189aa, 2951add5a59615f9a5c9012498e43bcc4893e3cbf0c7c778be00199d172e1fde6 c46676727e7fc164ef36d36b4536462c9b69ca9274dfd2722cef00ac76e76c4

jsbn/sec_mod.js by spruegel. Table 4.1 shows the value of each type of elliptic curve function.

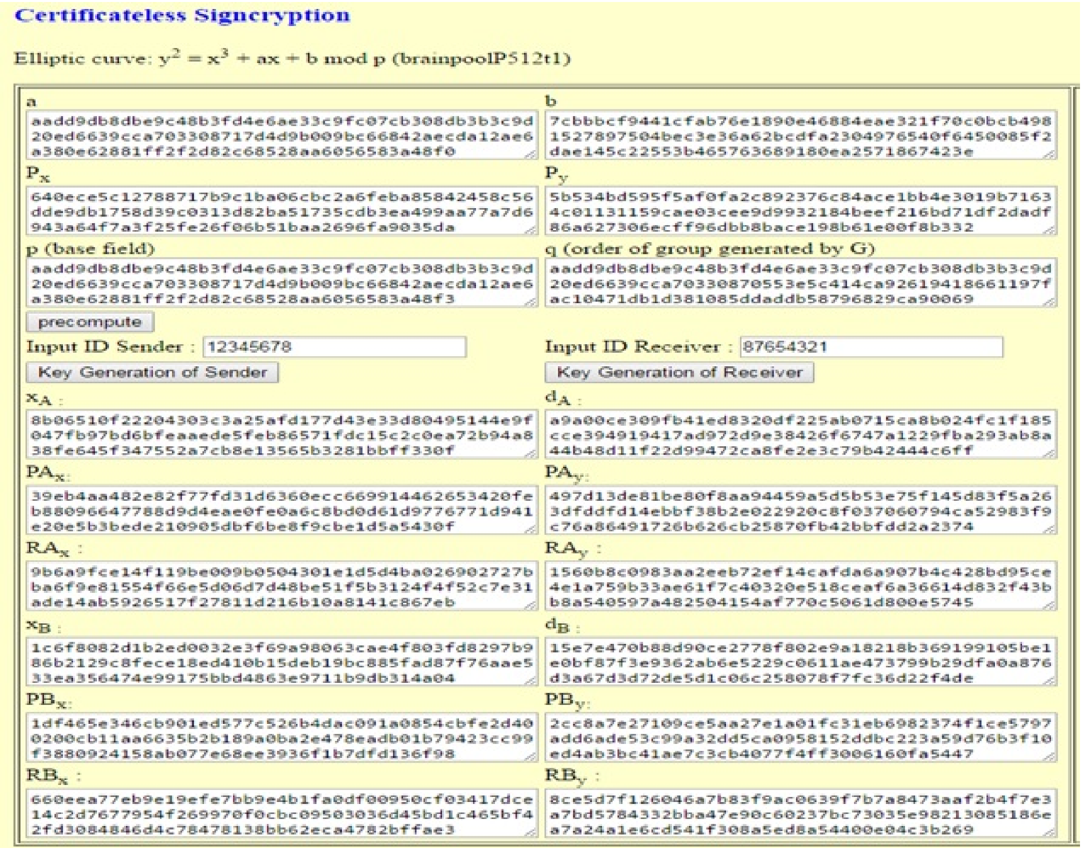
As for the operating point on its elliptic curve, we used a modification of the ec.js file belonging to Tom Wu. To increase the speed, we replaced the multiplication point simultaneously with faster-windowed method approach. In addition, for the speed in the processing of the extract key generation in this simulation, we made a key generation() function in the different js file. This file stores the master secret key msk owned by KGC that must remain confidential. The following figure is the functions that have been mentioned.



```
var extract = {
 keygeneration: function (ID,x,r)
        {
                var ecparams = getSECCurveByName(usedCurve);
                var q = ecparams.getN();
                var w = 4;
                var curve = get_curve();
                var P = new ECPointFp(curve);
                P = ecpComb2Mult(x,PA,w,GLOBAL_precomp,q.bitLength());
                var R = new ECPointFp(curve);
                R = ecpComb2Mult(r,RA,w,GLOBAL_precomp,q.bitLength());
                var hash = Crypto.SHA256(ID .concat (RA.getX ())
                        .concat(R.getY()).concat(P.getY()).concat(P.getY()));
                var hash1 = Crypto.util.hexToBytes(hash);
                var e = BigInteger.fromByteArrayUnsigned(hash1).mod(q);
                var msk = new BigInteger("3aea7fa0202e5d35038356102a6a
                        9a19eb114d94f56498da40849f4105a9016", 16);
                var x = e.multiply(msk).mod(q);
                var sk = r.add(x).mod(q).toString(16);
                return [sk,R,P];
        },
}
```

Figure 4.2: Key Generation function in javascript

For the one-way hash function, we have used SHA256 which takes the input of any length and then generates a 256-bit output. The symmetric encryption function used in the signcryption process is the AES algorithm. We took both of these functions from cryptoJS, JavaScript implementations of standard and secure cryptographic algorithms from https://code.google.com/p/crypto-js/. Figure 4.3 and figure 4.4 show the snapshot of our certificateless signcryption output.



Certificateless Signcryption

Elliptic curve: $y^2 = x^3 + ax + b \bmod p$ (brainpoolP512t1)

a

aadd9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca703308717d4d9b009bc66842aecda12ae6a380e62881ff2f2d82c68528aa6056583a48f0

b

7cbbbcf9441cfab76e1890e46884eae321f70c0bcb4981527897504bec3e36a62bcdfa2304976540f6450085f2dae145c22553b465763689180ea2571867423e

P_x

640ece5c12788717b9c1ba06cbc2a6feba85842458c56dde9db1758d39c0313d82ba51735cdb3ea499aa77a7d6943a64f7a3f25fe26f06b51baa2696fa9035da

P_y

5b534bd595f5af0fa2c892376c84ace1bb4e3019b71634c01131159cae03cee9d9932184beef216bd71df2dadf86a627306ecff96dbb8bace198b61e00f8b332

p (base field)

aadd9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca703308717d4d9b009bc66842aecda12ae6a380e62881ff2f2d82c68528aa6056583a48f3

q (order of group generated by G)

aadd9db8dbe9c48b3fd4e6ae33c9fc07cb308db3b3c9d20ed6639cca70330870553e5c414ca92619418661197fac10471db1d381085ddaddb58796829ca90069

Input ID Sender : 12345678

Input ID Receiver : 87654321

x_A :

8b06510f22204303c3a25afd177d43e33d80495144e9f047fb97bd6bfeaaede5feb86571fdc15c2c0ea72b94a838fe645f347552a7cb8e13565b3281bbff330f

d_A :

a9a00ce309fb41ed8320df225ab0715ca8b024fc1f185cce394919417ad972d9e38426f6747a1229fba293ab8a44b48d11f22d99472ca8fe2e3c79b42444c6ff

PA_x:

39eb4aa482e82f77fd31d6360ecc669914462653420feb88096647788d9d4eae0fe0a6c8bd0d61d9776771d941e20e5b3bede210905dbf6be8f9cbe1d5a5430f

PA_y:

497d13de81be80f8aa94459a5d5b53e75f145d83f5a263dfddfd14ebbf38b2e022920c8f037060794ca52983f9c76a86491726b626cb25870fb42bbfdd2a2374

RA_x :

9b6a9fce14f119be009b0504301e1d5d4ba026902727bba6f9e81554f66e5d06d7d48be51f5b3124f4f52c7e31ade14ab5926517f27811d216b10a8141c867eb

RA_y :

1560b8c0983aa2eeb72ef14cafda6a907b4c428bd95ce4e1a759b33ae61f7c40320e518ceaf6a36614d832f43bb8a540597a482504154af770c5061d800e5745

x_B :

1c6f8082d1b2ed0032e3f69a98063cae4f803fd8297b986b2129c8fece18ed410b15deb19bc885fad87f76aae533ea356474e99175bbd4863e9711b9db314a04

d_B :

15e7e470b88d90ce2778f802e9a18218b369199105be1e0bf87f3e9362ab6e5229c0611ae473799b29dfa0a876d3a67d3d72de5d1c06c258078f7fc36d22f4de

PB_x:

1df465e346cb901ed577c526b4dac091a0854cbfe2d400200cb11aa6635b2b189a0ba2e478eadb01b79423cc99f3880924158ab077e68ee3936f1b7dfd136f98

PB_y:

2cc8a7e27109ce5aa27e1a01fc31eb6982374f1ce5797add6ade53c99a32dd5ca0958152ddbc223a59d76b3f10ed4ab3bc41ae7c3cb4077f4ff3006160fa5447

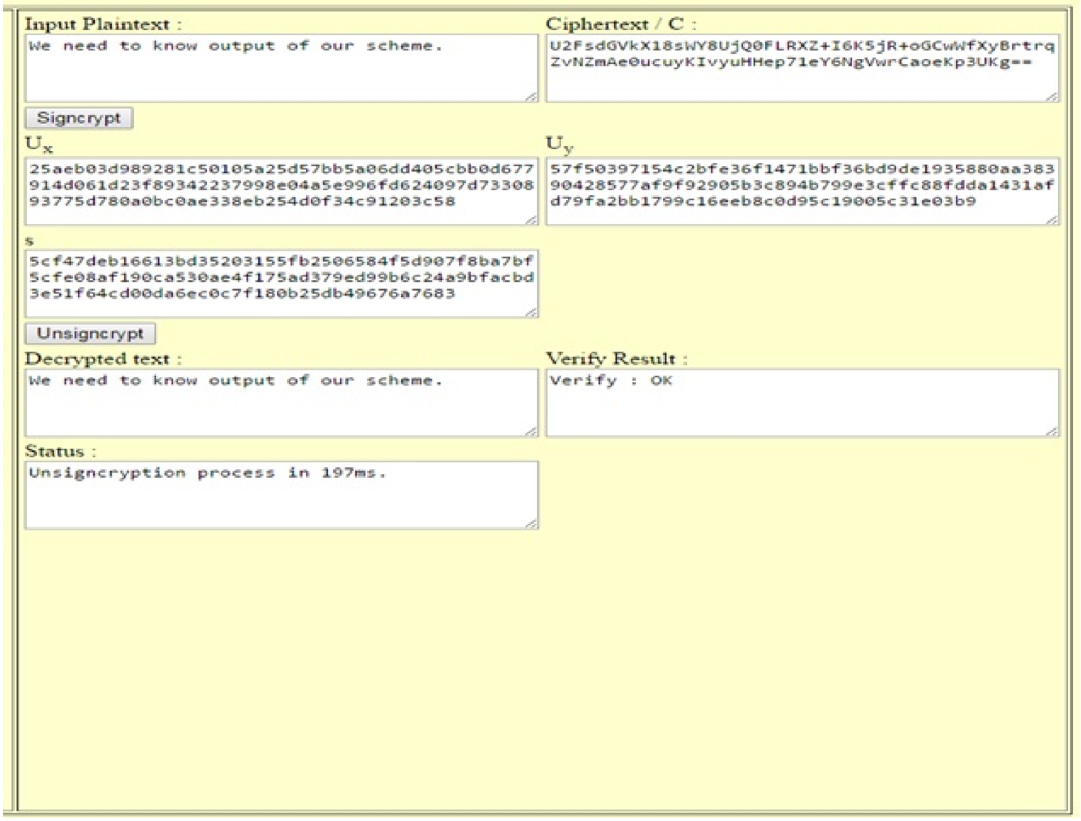
RB_x :

660eea77eb9e19efe7bb9e4b1fa0df00950cf03417dce14c2d7677954f269970f0cbc09503036d45bd1c465bf42fd3084846d4c78478138bb62eca4782bffae3

RB_y :

8ce5d7f126046a7b83f9ac0639f7b7a8473aaf2b4f7e3a7bd5784332bba47e90c60237bc73035e98213085186ea7a24a1e6cd541f308a5ed8a54400e04c3b269

Figure 4.3: Snapshot of Key Generation Result



Input Plaintext :

```
We need to know output of our scheme.
```

Ciphertext / C :

```
U2FsdGVkX18sWY8UjQ0FLRXZ+I6K5jR+oGCwWfXyBrtrq
ZvNZmAe0ucuyKIvyuHHep71eY6NgVwrCaoeKp3UKg==
```

U_x

```
25aeb03d989281c50105a25d57bb5a06dd405cbb0d677
914d061d23f89342237998e04a5e996fd624097d73308
93775d780a0bc0ae338eb254d0f34c91203c58
```

U_y

```
57f50397154c2bfe36f1471bbf36bd9de1935880aa383
90428577af9f92905b3c894b799e3cffc88fdda1431af
d79fa2bb1799c16eeb8c0d95c19005c31e03b9
```

s

```
5cf47deb16613bd35203155fb2506584f5d907f8ba7bf
5cfe08af190ca530ae4f175ad379ed99b6c24a9bfacbd
3e51f64cd00da6ec0c7f180b25db49676a7683
```

Decrypted text :

```
We need to know output of our scheme.
```

Verify Result :

```
Verify : OK
```

Status :

```
Unsigncryption process in 197ms.
```

Figure 4.4: Snapshot of Signcryption and Unsigncryption Result

4.5 Analysis of the Proposed CLSC

In this section, we evaluate the correctness of proposed certificateless signcryption scheme. Furthermore, we present a brief discussion about the security aspects of the proposed scheme. In addition, we offer the efficiency analysis in computational cost and speed performance after implementation.

4.5.1 Formula Correctness

The equation $SK = h2(l_A(Y_B + P_B), U, ID_B, R_B, P_B)$ in the Signcryption side and $SK = h2((d_B + x_B).U, U, ID_B, R_B, P_B)$ in the Unsigncryption side should be same. So equation $l_A.(Y_B + P_B)$ should be same with equation $(d_B + x_B).U$.

$$\begin{array}{lrl}
\texttt{While} & U = & l_A.P \\
 & d_B = & r_B + msk.h0(ID_B, R_B, P_B) \\
 & Y_B = & R_B + h0(ID_B, R_B, P_B).P_{pub} \\
\texttt{Then,} & (dB + xB).U & = rB + msk.h0(IDB, RB, PB) + xB).lA.P \\
 & & = l_A(r_B.P + msk.h0(ID_B, R_B, P_B).P + x_B.P) \\
 & & = l_A(R_B + h0(ID_B, R_B, P_B).P_{pub} + P_B) \\
 & & = l_A(Y_B + P_B)
\end{array}$$

Then, for the formula $s.P = Y_A + U.h1(m, ID_A) + P_A.h1(m, ID_A)$ should be hold. It is because,

$$\begin{array}{lll}
\texttt{While} & d_A.P & = (r_A + msk.h0(ID_A, R_A, P_A)).P \\
 & & = r_A.P + h0(ID_A, R_A, P_A).msk.P \\
 & & = R_A + h0(ID_A, R_A, P_A).P_{pub} = Y_A \\
\texttt{Then,} & s.P & = (d_A + l_A.h1(m, ID_A) + x_A.h1(m.ID_A)).P \\
 & & = d_A.P + l_A.P.h1(m, ID_A) + x_A.P.h1(m.ID_A) \\
 & & = Y_A + U.h1(m, ID_A) + P_A.h1(m, ID_A)
\end{array}$$

4.5.2 Security Analysis

A pair public and private key security rely on elliptic curve logarithm problem (ECLP). Partial public key $Y_A = R_A + H(ID_A, R_A, P_A).P_{pub} = d_A.P$, where $P_A, R_A, P_{pu}b$ and P is a point on the elliptic curve over a finite field and d_A is a quite large integer value. If partial private key d_A and P are given, it will be easy to compute partial public key Y_A. However if the ones given are partial public key Y_A and P, it will be hard to find partial private key d_A. On the same way for the other public key P_A, it comes from a random secret value $x_A(P_A = x_A.P)$. If secret value x_A and P are given, it will be easy to compute public key P_A. However if the ones given are public key P_A and P, it will be hard to find secret value x_A.

- **Confidentiality**.
  If the attacker tries to obtain the original message from the ciphertext, he has to know the keys SK. There are two ways to obtain the key by the attacker:

  $SK = h2(l_A.(Y_B + P_B), U, ID_B, R_B, P_B)$.

  If the attacker tries to derive the key SK from the above equation, he has to find out the l_A random value. In this case, it is infeasible to solve the SK key value from the above equation because l_A is obtained randomly and only used once.

  $SK = h2((d_B + x_B).U, U, ID_B, R_B, P_B)$

  It is also possible to derive the key from the above equation. However, the attacker has to obtain d_B and x_B because they are required to obtain SK. It is nearly impossible to obtain the SK from this second equation because d_B and x_B are the receivers private key which is known only by the receiver. In addition, because of the hardness of ECDLP, it is difficult to calculate x_A and dA from the equation $P_A = x_A.P$ and $Y_A = d_A.P$ even if P_A, Y_A and P are known.

- **Authentication**.
  User authentication
  The receiver uses the senders identity and public key (ID_A, R_A, P_A) and received digital signature (s) to verify sender authentication. The sender signs in with their private key (d_A, x_A). So here, the receiver can authenticate the identity of the sender.
  Data authentication
  The sender signs data m with her private key (d_A, x_A),
  $s = d_A + l_A.h_1(m, ID_A) + x_A.h_1(m, ID_A).mod\ q$ and send it to receiver. Then the receiver verifies the received data (m) using the received signature s. If $s.P = Y_A + U.h1(m, ID_A) + P_A.h1(m, ID_A)$ is hold, it means that the data actually comes from the true sender.

- **Integrity**.
  The receiver can verify whether the ciphertext was tampered or not at the time of transmission using the following equation.

  $s.P = Y_A + U.h1(m, ID_A) + P_A.h1(m, ID_A)$

  If the attacker changed the ciphertext c to c_1, then the received original message should also change from m to m_1. As a result, during verification, the computed digital signature of m_1 will not be the same as the digital signature of $m(s)$ sent by the sender to the receiver. Therefore, this scheme provides integrity.

- **Unforgeability**.
  Unforgeability ensures that the attacker cannot create a valid ciphertext. Here, the attacker cannot create a valid $C, U, s)$ without the private key of the sender. If an attacker forged a valid C', U', s' from the previous C, $U, s)$, then C', U', s' has to satisfy the
  $SK = h2(l_A(Y_B + P_B), U, ID_B, R_B, P_B)$ equation
  and $s = (d_A + l_A.h1(m, ID_A) + x_A.h1(m, ID_A)).mod\ q$. It is impossible to achieve without knowing l_A, d_A and x_A.

- **Non-repudiation**.
  In this scheme, the receiver knows from the
  $SK = h2(l_A(Y_B + P_B), U, ID_B, R_B, P_B)$ equation whether the original message was sent by the sender or not. The receiver can verify because the sender signs with his private key in the
  $s = (d_A + l_A.h1(m, ID_A) + x_A.h1(m, ID_A)).mod\ q$ equation. Thus it provides non-repudiation.

- **Forward Secrecy**.
  This scheme ensures that even though the senders private key is obtained. The attacker cannot recover original message m from the ciphertext C, $U, s)$. If the attacker tries to derive plaintext m, he must decrypt its ciphertext using secret key SK using random value l_A or secret key of the receiver. Therefore, our scheme provides forward secrecy.

Table 4.2 gives a comparison of security attributes and features between our proposed protocol scheme and others scheme.

4.5.3 Computational Cost Analysis

In this section, the time complexity of the proposed scheme is evaluated. Table 4.3 gives a comparison between the computational costs of our proposed scheme and those of the others schemes, in which the computational costs of verification and symmetric encryption are neglected. We used some notation to define a number of operation in that table which are given below.

- Exp = modular exponentiation operation
- Div = modular division operation

Table 4.2: Comparison of security properties of certificateless signcryption schemes and their variants

Schemes	**Conf.**	**Auth.**	**Int.**	**UF.**	**Non-Repud.**	**Forwd.Secrecy**
Zheng[26]	Yes	Yes	Yes	Yes	No	No
ZI[36]	Yes	Yes	Yes	Yes	No	No
WNPZ[32]	Yes	Yes	Yes	Yes	Yes	Yes
XX[31]	Yes	Yes	Yes	Yes	Yes	Yes
SB[28]	Yes	Yes	Yes	Yes	Yes	unknown
WSB[29]	Yes	Yes	Yes	Yes	Yes	unknown
Ours	Yes	Yes	Yes	Yes	Yes	Yes

Yes/No: Feature shown in the left column is/is not held.
unknown: It is unknown whether the security property shown in the top is achieved.
Conf.:Confidentiality, Auth.:Authentication, UF.:Unforgeability, Non-Repud.:Non-Repudiation, Forwd.Secrecy:Forward Secrecy.

- Mul = modular multiplication operation
- Add = modular addition operation
- ECMult = Elliptic Curve point multiplication operation
- ECAdd = Elliptic Curve point addition operation
- Hash = One way hash function

Besides the computational cost based on the mathematic operation, we evaluate the performance of several processes of our certificateless signcryption scheme by implementation. We make comparison of our scheme with only two existing CLSC schemes (SB[28] and WSB[29]) which are also based on elliptic curve. We do not compare our scheme with the other existing schemes because they use bilinear pairings or finite field exponentiation technique which have slower computation than elliptic curve computation. Figure 4.5 shows performances in execution time of each scheme in milisecond while table 4.4 shows the comparison of the size of ciphertexts transmitted from sender to receiver.

Based on the figure 4.5, we can see that our protocol performance is faster than SB[28] scheme and almost same speed with WSB[29] scheme. In table 4.4, we can see that our protocol has a shorter ciphertext size than WSB[29] scheme and same with SB[28] scheme. It means that the performance of our protocol is better than two other schemes.

Table 4.3: Computational costs of different schemes

Schemes	Type	Participant	Exp	Div	Mul	Add	ECMult	ECAdd	Hash
Zheng[26]	SC	Sender	1	1	-	1	-	-	2
		Receiver	2	-	2	-	-	-	2
ZI[36]	SC	Sender	-	1	1	1	1	-	2
		Receiver	-	-	2	-	2	1	2
		Receiver	-	-	-	-	4	2	2
WNPZ[32]	CLSC	Sender	4	1	3	2	-	-	4
		Receiver	5	-	3	2	-	-	4
XX[31]	CLSC	Sender	5	-	4	2	-	-	3
		Receiver	5	-	4	2	-	-	3
SB[28]	CLSC-TKEM	Sender	-	-	3	2	4	1	4
		Receiver	-	-	-	-	6	3	4
WSB[29]	CLSC-TKEM	Sender	-	-	2	2	4	2	4
		Receiver	-	-	-	1	6	3	4
Ours	CLSC	Sender	-	-	2	2	3	2	3
		Receiver	-	-	-	1	5	3	3

SC: Signcryption, CLSC: Certificateless Signcryption, CLSC-TKEM: Certificateless Signcryption-Tag Key Encapsulation Mechanism.

We executed using windows 64-bit operating system, processor intel(R) Core(TM) i7-3770 CPU @3.40 GHz and memory (RAM) 16.0 GB. From that tables, we can see that bit length of the used elliptic curve influences of the speed of the system because has relation to the mathematic operation. While the Identity length does not effect to the speed. It is because one-way hash function will execute in the same way from different input length to produce same output length.

Table 4.4: Ciphertext size comparison

EC-CLSC Schemes	Ciphertext Size
SB[28]	$n_q + n_G + n_{ID} + m$
WSB[29]	$n_q + 2n_G + n_{ID} + m$
Ours	$n_q + n_G + n_{ID} + m$

n_q: The number of bits required to represent an element of F_q
n_G: The number of bits required to represent an element of point EC
n_{ID}: The number of bits required to represent an identity
m: The number of bits in the message being signcrypted.

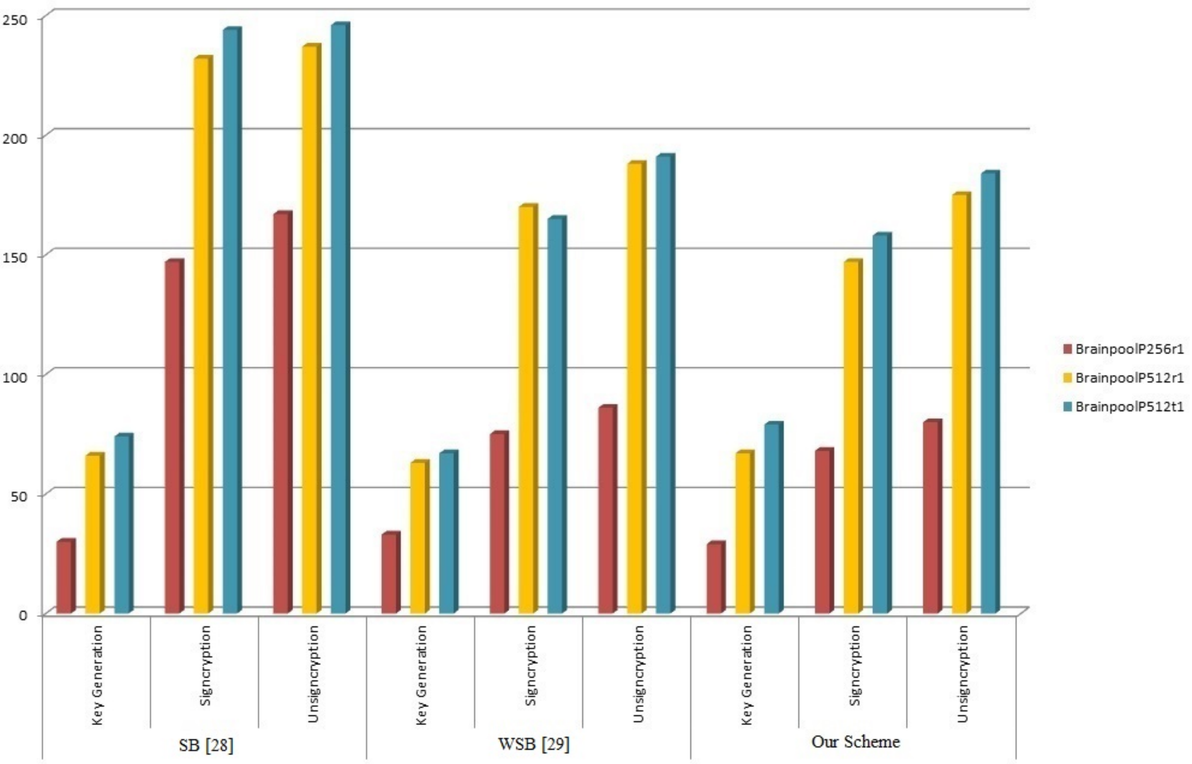


Figure 4.5: Comparison of performance of the CLSC schemes based on elliptic curve

Chapter 5

Secure Online Examination Protocol

The security protocols of the Online Examination Systems (OES) has differences with the security protocols of other systems. If in the security protocol of other systems we only need to consider a security issues, in the OES protocols we have to consider its own security issues and it also has to pay attention to its cheating problems. Existing security protocols such as secure http (https) cannot be used to solve all problems in the OES. Therefore, we need a particular security protocol that is specifically for use in OES.

In this chapter, we offer a secure online examination protocol based on certificateless signcryption scheme. We adopt a certificateless signcryption protocol that we have designed in the previous chapter with a slight modification. In order to prove the security of our OES protocol, we model our OES protocol using an applied pi-calculus and then prove its security by formal analysis. We use ProVerif software for this purpose.

5.1 Background

Online Examination System (OES) has been developed as an alternative method for test execution system. Although this system has not been able to replace conventional test systems using paper, the system is expected to be one of the alternative methods for test execution system. OES is supported by the Question Bank information system that provides valid question data with the multiple-choice format (objective) which has been stored in the question bank database. Trials were conducted in special rooms such as ICT centers or computer laboratories with a network system that is spread throughout Indonesia or the entire campus.

EOS was developed by taking advantage of internet technology for the data communication process. Thus, a proper mechanism is required to safeguard the security of data on the OES, especially when there is communication involving Internet network where the data is prone to be stolen, altered or destroyed. This mechanism is commonly known as the Protocol, in which the protocol that is widely used nowadays for data security system is a cryptographic protocol. The cryptographic protocol is a protocol that uses a cryptographic system. People who participate in the cryptographic protocol need it to share the secret component in calculating a value, generate a series of random numbers, authentication and so forth. Cryptographic protocols are built by engaging multiple cryptographic algorithms which are mostly designed to be used by groups consisting of two users, but some are also designed to be worn by groups consisting of more than two users.

Best on our knowledge, there has not been researching that specifically addresses the protocol used in OES. There are at least three papers that have specifically discussed the OESs security protocol. The first paper is from Castella et.,al. [12], in their paper, they attempt to obtain a secure e-exam management system with some cryptographic primitive methods such as encryption and digital signatures. The protocol is based on a trusted third party (TTP), manager. Besides, this protocol system is protected using conventional multiple security features such as firewalls, VPN, IDS, etc. The system also uses Public Key infrastructure (PKI) for certificate management.

The second paper is from Huszty and Petho [39]. They design a security protocol that is almost similar to the scheme by Castella et.al. It uses cryptographic primitive models but tries to eliminate the trusted manager (TTP) for the sake of anonymity property. The anonymity property is accomplished by utilizing pseudonyms issued by timed-released service by some of servers. But unfortunately, it has been found some security flaw in this scheme [40].

Rosario et.,al. [41] tried to construct Remark! in the third paper. In this paper, they are written that Remark! achieves authentication, verifiability, and conditional anonymity with minimal reliance on trusted parties. They use Mix-net exponentiation method to obtain anonymity. As a result, this protocol uses multiple server so that they have a problem of computational cost and time overhead. In additional, It is same of two previous protocol that uses PKI and it cause some problems in the certificate management such as how to design certificates authority, how to handle revocation user and how to manage a key.

In our scheme, we try to address the certificate problems which explained in previous chapter, by adopting certificateless cryptography and then, in order to increase the efficiency, signcryption could be an alternative solution to replace the classical method signature-then-encryption.

The objective of this chapter is to design a security protocol for OES which is more efficient than currently existing protocols for specific OES system with keeping its security.

5.2 OES Basic Assumptions and Network Architecture

For the sake of our convenience in the design, some basic assumptions used as an initial requirement for the exam, are as follows:

1. The examinee will use a smart card that stores the public key and private key which are generated after the initial registration by filling in the biographical data requested by the registrations server.

2. The smart card must also contain the identity and the examinees photo as authentic proof which shows that the examinee who holds the smart card is the actual owner.

3. The exam is only done at a certain time and location, according to a schedule determined when performing the initial registration and has been published earlier.

4. The PC that can be used during the exam is final and its position cannot be exchanged, so that it will reduce the possibility of the arrangement of the examinees seating position which may make cheating possible.

5. The system is physically located in a secure place such as a computer lab or ICT center and has already been set for OES.

6. There is control requirement such login system procedure to access public board by examinees and examiners.

In general, the OES data communications system is a client-server relationship, because it involves the only computer on the side of the examinee as the client and OES service and application provider computer as the computer server. Client and server are connected by an Internet network. Each communication process between the client and the server will involve important data which needs to be safeguarded. Some of the data include:

1. Online Exam Setup Data. When performing the initial setup by sending an examiners or examinee's identity (user id) for example email or

ID Number and his public value to the Key Generation Centre (KGC), the KGC will then generate a partial public key and a partials private key. Delivery of the private key from the KGC to the examiner or the examinee must be secure. We use a smart card to store it and to be given directly to the examinee.

2. Examination Data. Security of the exam questions data from its storage aspect as well as from the delivery aspect is the next critical point in our security protocol design. The questions data must be secured when inputted by the Examiners, sent until it is stored in the database. Similarly, during the examination process starting from the retrieval from the database to the delivery process from the server to the client is a critical point in the security protocol that must be considered carefully, so that the data cannot be tampered with, altered, modified or stolen.

3. Exam Answers Results Data. Answers produced by the examinee must be sent back to the OES server for the assessment process. Delivery of the exam answer results of the examinee/client to the server is another critical point which may become a security flaw in the protocol that requires attention.

4. Exam Marks. After the exam answers are checked and assessed, the assessment results data need to be sent back from the examiner to the examinee and also securely stored in the database. The safety of the process of sending and storing marks data must also be ensured.

5.3 Threats and Security Properties

A threat is a potential force in a security system. In the context of the exam, there are many threatening attacks. Therefore, the test system to be built must be prepared to be able to protect systems from the threats that may occur.

Threats that may occur in the conventional exams include question papers received by the examinees are wrong, leaked questions paper or the use of someone else's identity to take the exam. Wrong and leaked question papers may occur as a result of the process of preparing, printing and distributing the question papers that require a long time. Such threats may occur due to intrusion by unauthorized individuals or the modification of questions by intruders. The threat of the use of another person's identity occurs as a result of poor identification process and surveillance.

Threatening attacks on conventional exams are different from the attacks on the OES. Threats that occur on the OES may occur in the process of exchanging information which is sent and received by the client and the server. Data transmission security protocols on OES has the potential to receive threats. An analysis of various threats that may emerge and attack security protocol is highly required. In OES, there are some objects that could potentially be targeted, among others: the session key, registration data, exam question papers, exam answers and exam results. The threats that are expected to emerge, among others are:

1. Spoofing that occurs during the process of communication between the client and the server, such as when the client conducts registration, login or activation for exams or also at the time of sending the exam answers or results. Incognito also allows the repudiation of data sent by a particular party. In this case, one party could deny sending the data to other parties because the identity has been changed by the attacker. This threat will be overcome by the authentication service.

2. Interception, in this case, what may occur is the intruder managed to read the registration data, session key, and exam questions. This threat can be overcome by privacy or confidentiality services.

3. Modification, in this case, the threat of modifications that may occur in which the attacker changes the symmetric key or session key that is used in the communication process between the client and the server. This can cause the communication to be disrupted or even cannot be done. Another possible threat is the attacker changes the questions or exam answer results that were sent. This threat will be overcome with integrity service.

A security protocol must meet the basic needs of security in accordance with the required cryptographic aspects. Cryptographic aspects have become an inevitable necessity in an electronic transaction like OES. Some of the aspects that should become the security property of a security protocol of an OES are as follows:

1. **Confidentiality**. All data related to the OES which includes exam questions, answers to the exam as well as test results must be kept confidential. Confidentiality service is done by applying a cryptographic protocol which is encryption. Some of confidentiality required in this OESs protocol include:

   - Question Confidentiality

- Answer Secrecy
- Mark Secrecy.

2. **Authentication**. In this case, the authentication requirements in OES are entity authentication and data authentication. Entities authentication ensures that all examinee, managers, examiners entity are all correct entities. While data authentication ensures that all data in the OES process is the correct data coming from the corrects sender.

   - Examiner Authentication
   - Examinee Authentication.

3. **Data integrity**. All data contained in OES process must be guaranteed to be in an intact condition and unchanged. Data integrity is guaranteed by using a digital signature on all of the sent data.

   - Question original
   - Answer original
   - Mark original.

4. **Non-repudiation**. Both client and server cannot deny that they had sent the data. This requirement is guaranteed by using digital signatures of each party.

   - Question Authenticity
   - Answer Authenticity
   - Mark Authenticity

5.4 Our Proposed OES Protocol

5.4.1 Notation

In our scheme, we consider to three participants which are Examiner, Manager, and Examinee. Besides we have a trusted center, KGC (Key Generation Centre). A public board is used by Manager to publish the Exam question forms, the Answer forms, and the Marks. It is also used by KGC to publish public key of all participants, Examiners, Manager, and Examinees. In this proposed protocol, we just explain a system by manually grading because a system by automatic grading will simpler. The following notations are used in order to describe our protocol.

- KGC: Symbol of Key Generation Centre.
- A: Symbol of Examiner.
- M: Symbol of Manager.
- C: Symbol of Examinee.
- $(ID_{[participant]})$: Identity of the participant.
- $(d_{[participant]}, x_{[participant]})$: a certificateless secret key pair of the participant which consist of a Partial private key and Secret Value.
- $(r_{[participant]})$: Random secret value of participant which is created by KGC.
- $(R_{[participant]}, P_{[participant]})$: a certificateless public key pair of the participant which is constructed from random secret value r and secret value x respectively.
- $Y_{[participant]}$: the Public key which is constructed from partial private key $d_{[participant]}$.
- P: Basepoint in the elliptic curve which is used in the protocol.
- l_j, U_j: Pair of random value and multiplication of random value to point P respectively which are used to construct of certificateless signcryption. Subindex j identifies the number of the pair in this protocol.
- (C_j, U_j, s_j): certificateless signcryption result. Subindex j identifies the number of signcryption in this protocol.
- s_j: certificateless signature result. Subindex j identifies the number of signature in this protocol.
- SK_j: A session key from the point of the elliptic curve. Subindex j identifies the number of key in this protocol.
- $C = E(m, SK_j)$: symmetric key encryption of message m, using key SK_j.
- $m = D(C, SK_j)$: symmetric key decryption of C, using key SK_j.
- $h0, h1, h2$: Hash function.

5.4.2 Set-Up of System

1. Setup Preparation, it is run by KGC:
   Set-Up Parameter: Here, KGC will first generate and select several security parameters to be used as a parameter when building a secure communications protocol. The security parameters are same with securities parameter in chapter 4. KGC publishes the public parameters $(q, a, b, E/F_q, P, h0, h1, h2, P_{pub})$ but keeps the msk secret.

2. Set secret value, it is run by each participant: User X with Identity ID_X

   - Choose at random $x_X \in Z_q^*$.
   - Computes corresponding Public key $P_X = x_X.P$.
   - User X then sends ID_X and P_X to KGC to get her partial private key.

3. Partial private key extract, it is run by KGC: KGC receives ID_X and P_X from user X and starts to run the following actions.

   - Choose at random $r_X \in Z_q^*$
   - Compute $R_X = r_X.P$.
   - After that, compute $d_X = r_X + msk.h0(ID_X, R_X, P_X).mod\ q$.
   - KGC then sends a partial private key d_X to user X in a secure channel and publishes R_X, P_X which bounded to his identity ID_X in the public board.

4. Each user then sets their private key pair (d_A, x_A) for examiner, (d_M, x_M) for manager and (d_C, x_C) for examinee and sets their public key pair (R_A, P_A) for examiner, (R_M, P_M) for manager and (R_C, P_C) for examinee.

5.4.3 Set-Up of an Exam Question

1. Examiner ID_A want to send an exam test form $(idex, question)$ to Manager ID_M. Examiner performs the following actions:

   - Chooses $l_1 \in Z_q^*$, and compute $U_1 = l_1.P$.
   - Computes $Y_M = R_M + h0(ID_M, R_M, P_M).P_{pub}$.
   - Computes a session key for encrypt between Examiner and Manager, $SK_1 = h2(l_1.(Y_M + P_M), U_1, ID_M, R_M, P_M)$.

- Encrypts $idex, question$ which is bounded to ID_A,
  $C1 = E((idex, question, ID_A), SK_1)$.
- Computes digital signature of $idex$, $question$ and ID_A:
  $s_1 = d_A + l_1.h1(idex, question, ID_A) + x_A.h1(idex, question, ID_A)$.
- Sends (C_1, U_1, s_1) to the manager.

2. Manager (ID_M) receives (C_1, U_1, s_1) from Examiner (ID_A). Manager then performs the following actions:

   - Computes a session key for encrypt between Examiner and Manager, $SK_1 = h2((d_M + x_M).U_1, U_1, ID_M, R_M, P_M)$.
   - Decrypt C_1 using SK_1 to obtain $idex, question, ID_A$.
     $(idex, question, ID_A) = D(C_1, SK_1)$.
   - Computes $Y_A = R_A + h0(ID_A, R_A, P_A).P_{pub}$.
   - Verifies the signcryption with computes: $s_1.P = Y_A + h1(idex, question, ID_A).U_1 + h1(idex, question, ID_A).P_A$.
     If verification is true then the Manager stores an exam test form $(idex, question)$ which is bounded to the ID_A in a secure way.
   - After that, Manager executes several steps below: Chooses $l_2 \in Z_q^*$, and compute: $U_2 = l_2.P$. Computes signatures:
     $s_2 = d_M + l_2.h1(idex, question, ID_A) + x_M.h1(idex, question, ID_A)$.
   - Publishes (U_2, s_2, ID_A) to public board as a receipt that he has received her exam test form $(idex, question)$.

3. Examiner (ID_A) gets (U_2, s_2) from public board and verifies signature with computes:
   $s_2.P = Y_M + h1(idex, question, ID_A).U_2 + h1(idex, question, ID_A).P_M$

5.4.4 Testing Process

1. Manager (ID_M) wants to distribute an exam test form (ID_M)(idex, question) which is bounded with ID_A to Examinee (ID_C) to the public board. We assume that each examinee has been authenticated by Manager and only authenticated examinee can access the public board. For each Examinee, the manager makes the following actions :

   - Chooses $l_3 \in Z_q^*$, and computes $U_3 = l_3.P$.
   - Computes $Y_C = R_C + h0(ID_C, R_C, P_C).P_{pub}$.
   - Computes a session key for encryption key between Manager and Examinee, $SK_2 = h2(l_3.(Y_C + P_C), U_3, ID_C, R_C, P_C)$.

- Encrypts $idex, question, ID_A$ and ID_M,
  $C2 = E((idex, question, ID_A, ID_M), SK_2)$.
- Computes digital signature of $idex, question, ID_A$ and ID_M,
  $s_3 = d_M + l_3.h1(idex, question, ID_A, ID_M) + x_M.h1(idex, question, ID_A, ID_M)$.
- Sends (C_2, U_3, s_3) to the public board.

2. Examinee (ID_C) gets (C_2, U_3, s_3) from Manager (ID_M) in the public board and performs the following actions:

   - Computes $SK_2 = h2((d_C + x_C).U_3, U_3, ID_C, R_C, P_C)$.
   - Decrypts C_2 using SK_2 to obtain $idex, question, ID_A, ID_M$.
     $(idex, question, ID_A, ID_M) = D(C_2, SK_2)$.
   - Computes: $Y_M = R_M + h0(ID_M, R_M, P_M).P_{pub}$.
   - Verifies the signcryption with computes: $s_3.P = Y_M + h1(idex, question, ID_A, ID_M).U_3 + h1(idex, question, ID_A, ID_M).P_M$. If verification is true then examinee answers the question to produce answer form ($answer$) and then performs:
     - Chooses $l_4 \in Z_q^*,$, and compute $U_4 = l_4.P$.
     - Computes $Y_A = R_A + h0(ID_A, R_A, P_A).P_{pub}$.
     - Computes a session key agreement between Examiner and Examinee, $SK3 = h2(l_4.(Y_A + P_A), U_4, ID_A, R_A, P_A)$.
     - Chooses $l_5 \in Z_q^*$, and compute $U_5 = l_5.P$.
     - Computes signatures of answer form which is bounded with idex, $s_4 = d_C + l_5.h1(idex, answer) + x_C.h1(idex, answer)$.
     - Encrypts of $idex, answer$ and its signature s_4 using a session key agreement SK_3, $C_3 = E((idex, answer, s_4), SK_3)$.
     - Chooses $l_6 \in Z_q^*$, and compute $U_6 = l_6.P$.
     - Gets Y_M again.
     - Computes a session key for encryption between Examinee and Manager, $SK_4 = h2(l_6.(Y_M + P_M), U_6, ID_M, R_M, P_M)$.
     - Encrypts C_3, U_4, U_5, ID_A and ID_C:
       $C_4 = E((C_3, U_4, U_5, ID_A, ID_C), SK_4)$.
     - Computes digital signature of C_3, U_4, U_5, ID_A and (ID_C), $s_5 = d_C + l_6.h1(C_3, U_4, U_5, ID_A, ID_C) + x_C.h1(C_3, U_4, U_5, ID_A, ID_C)$.
     - Sends (C_4, U_6, s_5) to the manager.

3. Manager (ID_M) receives (C_4, U_6, s_5) from Examine (ID_C) and then performs the following actions:

- Computes $SK_4 = h2((d_M + x_M).U_6, U_6, ID_M, R_M, P_M)$.
- Decrypts (C_4) using (SK_4) to obtain C_3, U_4, U_5, ID_A and (ID_C). $(C_3, U_4, U_5, ID_A, ID_C) = D(C_4, SK_4)$.
- Gets (Y_C) from the session.
- Verifies the signcryption with computes: $s_5.P = Y_C + h1(C_3, U_4, U_5, ID_A, ID_C).U_6 + h1(C_3, U_4, U_5, ID_A, ID_C).P_M$, if verification is true then stores $(C_3, U_4, U_5, ID_A, ID_C)$ in a secure way and then:
  - Chooses $l_7 \in Z_q^*$, and compute $U_7 = l_7.P$.
  - Computes signatures of C_3 which is bounded with (ID_C), $s_6 = d_M + l_7.h1(C_3, ID_C) + x_M.h1(C_3, ID_C)$.
  - Publishes U_7, s_6, corresponding with ID_C to the public board as a receipt for the examinee.

4. Examinee ID_C gets (U_7, s_6) from public board and verifies signature and computes $s_6.P = Y_M + h1(C_3, ID_C).U_2 + h1(C_3, ID_C).P_M$

5.4.5 Marking Process

1. Manager (ID_M) needs to send 'answer form' of the examinee (ID_C) to related Examiner (ID_A) for marking process. He performs the following actions:

   - Creates new random idX to bound 'answer' which has a relationship with Examinee (ID_C).
   - Chooses $l_8 \in Z_q^*$, and compute $U_8 = l_8.P$.
   - Computes signature of (C_3, U_4) which is bounded with idX, $s7 = d_M + l_8.h1(C_3, U_4, idX) + x_M.h1(C_3, U_4, idX)$.
   - Publishes signature C_3, U_4, idX, U_8, s_7, which is bounded to (ID_A) to the public board.

2. Examiner (ID_A) gets C_3, U_4, idX, U_8, s_7 from Manager (ID_M) in the public board and starts to perform the following actions:

   - Verifies signatures with computes: $s_7.P = Y_M + h1(C_3, U_4, idX).U_8 + h1(C_3, U_4, idX).P_M$. If verification is true then:
     - Computes $SK_3 = h2((d_A + x_A).U_4, U_4, ID_A, R_A, P_A)$.
     - Gets (Y_A) again.

  - Decrypts (C_3) using (SK_3) to obtain $idex, answer$ and (s_4). ($idex, answer, s_4$) $= D(C_3, SK_3)$.
  - Examine start to evaluate the answer base on idex of the question and gives him a marking 'Mark'.
  - Then, chooses $l_9 \in Z_q^*$, and compute ($U_9 = l_9.P$.
  - Computes signatures of ($idex, Mark$) which is bounded with idX, $s_8 = d_A + l_9.h1(idex, Mark, idX) + x_A.h1(idex, Mark, idX)$.
  - Sends $idex, Mark, idX$, U_9, s_8 to the Manager (ID_M).

3. Manager (ID_M) receives signatures ($idex, Mark, idX$, U_9, s_8) and makes the following steps:

   - Verifies signatures with computes: $s_8.P = Y_A + h1(idex, Mark, idX).U_9 + h1(idex, Mark, idX).P_A$. If verification is true then Manager finds (ID_C) which has a relationship with idX and stores (ID_C, $idex, Mark$, U_5, idX) and (ID_A) in a secure way.
   - Then, chooses $l_{10} \in Z_q^*$, and compute $U_{10} = l_{10}.P$.
   - Computes signatures of ($Mark$) which is bounded with (ID_A), $s_9 = d_M + l_9.h1(Mark, ID_A) + x_M.h1(Mark, ID_A)$.
   - Publishes signatures U_{10}, s_9, ID_A to the public board as a receipt for the examiner.

4. Examiner (ID_A) gets (U_{10}, s_9) from public board and verifies signature with computes $s_9.P = Y_M + h1(Mark, ID_A).U_{10} + h1(Mark, ID_A).P_M$.

5.4.6 Notification Process

1. Manager (ID_M) starts to give notification of test result of Examinee (ID_C). Before manager gives notification to the examinee, he needs to confirm the original of 'answer' to Examiner (ID_A). Manager performs the following actions:

   - Chooses $l_{11} \in Z_q^*$, and computes $U_{11} = l_{11}.P$.
   - Gets (Y_A) again.
   - Computes a session key for encryption between Manager and Examiner, $SK_5 = h2(l_{11}.(Y_A + P_A), U_{11}, ID_A, R_A, P_A)$.
   - Encrypts $ID_C, Mark, U5, idX$ and (ID_M), $C_5 = E((ID_C, Mark, U_5, idX, ID_M), SK_5)$.

- Computes digital signature of $ID_C, Mark, U_5, idX$ and (ID_M), $s_{10} = d_M + l_{11}.h1(ID_C, Mark,$ $U_5, idX, ID_M) + x_M.h1(ID_C, Mark, U_5, idX, ID_M)$.
- Sends (C_5, U_{11}, s_{10}) which is bounded to (ID_A) to the public board.

2. Examiner (ID_A) receives (C_5, U_{11}, s_{10}) from the manager (ID_M) on the public board. He starts to check the origin of the 'answer' which he has marked in the previous session. Examiner performs the following actions:

   - Computes $SK_5 = h2((d_A + x_A).U_{11}, U_{11}, ID_A, R_A, P_A)$.
   - Decrypts (C_5) using (SK_5) to obtain $(ID_C, Mark, U_5, idXandID_M)$, $(ID_C, Mark, U_5, idX, ID_M) = D(C_5, SK_5)$.
   - Gets (Y_M) again.
   - Verifies the signcryption with computes: $s_{10}.P = Y_M + h1(ID_C,$ $Mark, U_5, idX, ID_M).U_{11} + h1(ID_C, Mark, U_5, idX, ID_M).P_M$, if verification is true then get $(idex, answer, s_4)$ from previous session using the same idX.
   - Verifies signatures with computes: $s_4.P = Y_C + h1(idex, answer).U_5 + h1(idex, answer).P_C$. If verification is true then:
     - Creates OK signal.
     - Chooses $l_{12} \in Z_q^*$, and compute $U_{12} = l_{12}.P$.
     - Computes signatures of (OK) signal, $s_{11} = d_A + l_{12}.h1(OK) + x_A.h1(OK))$.
     - Sends signatures U_{12}, s_{11} to the manager (ID_M).

3. Manager (ID_M) receives (U_{12}, s_{11}) from Examiner (ID_A) and verify signatures with computes $s_{11}.P = Y_A + h1(OK).U_{12} + h1(OK).P_A$. If verification is true then:

   - Chooses $l_{13} \in Z_q^*$, and compute $(U_{13} = l_{13}.P)$.
   - Gets (Y_C) again.
   - Computes a session key for encryption between Manager and Examinee, $SK_6 = h2(l_{13}.(Y_C + P_C), U_{13}, ID_C.R_C, P_C)$.
   - Encrypts $ID_C, idex$ and $Mark$, $C_6 = E((ID_C, idex, Mark, ID_M), SK_6)$.
   - Computes digital signature of $ID_C, idex$ and $Mark$, $s_{12} = d_M + l_{13}.$ $h1(ID_C, idex, Mark, ID_M) + x_M.h1(ID_C, idex, Mark, ID_M)$.

- Sends (C_6, U_{13}, s_{12}) which is bounded to (ID_C) to the public board.

4. Examinee (ID_C) receives (C_6, U_{13}, s_{12}) from manager (ID_M) in the public board and performs the following actions:

   - Computes a session key $SK_6 = h2((d_C + x_C).U_{13}, U_{13}, ID_C, R_C, P_C)$.
   - Decrypts C_6 using (SK_6) to obtain $(ID_C, idex, Mark$ and $ID_M)$, $(ID_C, idex, Mark, ID_M) = D(C_6, SK_6)$.
   - Gets (Y_M) again.
   - Verifies the signcryption with computes: $s_{12}.P = Y_M + h1(ID_C, idex, Mark, ID_M).U_6 + h1(ID_C, idex, Mark, ID_M).P_M$. If verification is true then Examinee (ID_C) get his result of the test.

5.5 Formal Analysis of Our Protocol

We can model the roles of online examination protocol as processes in the applied phi-calculus. All of these process will communicate using public or private channels, and can create a new random values, which can serve as a key or nonce, such as. The process performs tests and cryptographic operations, which are functions on terms with respect to a theory of equational describing some algebraic properties.

We analyze our protocol in ProVerif. We consider to privacy and authentication properties formally specified in previous section. The property with simple description are recalled below:

1. Privacy

   - **Question Confidentiality**, which says that the questions are not revealed until testing begins and by another persons.
   - **Answer Privacy**, which says that no one can reveal the answers of examinee besides the concerned examiner, include the manager.
   - **Mark Privacy**, which says that no one learns the marks, besides the examiner, the concerned examinee, and the manager (notifier).
   - **Mark Anonimity**, which says that no one learns the association between a mark and the corresponding candidate.

2. Authentication

   - **Examiner Authorisation**, which says that only registered examiner can submit questions to the server.

- **Examinee Authorisation**, which says that only registered examinee can take the exam.
- **Question Authenticity**, which says that the manager consider only the question that the examiner actually submitted.
- **Answer Authenticity**, which says that the manager consider only the answer that the examinee actual submitted.
- **Test Origin Authentication**, which say that the manager accept only answer that originate from registered examinee.
- **Test Authenticity**, which says that the examiner only marks the tests intended for him.
- **Mark Authenticity**, which says that the examinee receives the Mark assigned to her test by the examiner chosen by the manager.

In order to model privacy requirements as reachability properties, we just check some terms which are needed to achieve by reachability-based secrecy models, such as:

- **query attacker(question)**; This query checks that attackers can reach the term question or not.
- **query attacker(answer)**; This query checks that attackers can reach the term answer or not.
- **query attacker(Mark)**; This query checks that attackers can reach the term Mark or not.

Besides, we are necessary to define a number of relevant events in order to model authentication requirements as correspondence properties. Events normally need to agree with some arguments to capture authentication. Thus we introduce the terms that serve as arguments in our events as follow.

- ID_A refers to identity of Examiners.
- ID_C refers to identity of Examinee.
- ID_M refers to identity of Manager.
- *question* donates the questions of the exam.
- *answer* donates the answers of the exam.
- *Mark* donates the marks assigned to the exam.

- d_A, d_C, d_M refers to partial private key of Examiner, Examinee and Manager respectively

- *idex* refers to identifier of the exam

Then, we define a list of ten events that allow to specify six fundamental authentication requirements for exams. We stress that the list can be futher extended to accommodate any additional requirements.

- event reg(ID_X, d_X) means that the KGC considers the all participants registered for the exam. The events is inserted into the process of KGC at the location where the registration of participants concludes.

- event QuestionReceive(ID_A, *idex*, *question*) means that the manager accepts the test *idex*, which originates from the Examiner ID_A. The event is inserted into the process of manager at the location where setup exam question is considered as accepted.

- event AnswersReceive(ID_C, *C_3*, ID_A) means that the manager accepts the encrypted answer *C_3*, which originates from the examinee ID_C, associated with the examiner ID_A.The event is inserted into the process of manager at the location where answer from examinee is considered as accepted.

- event AnswerSubmit(ID_C, *idex*, *question*, *answer*, ID_A) means that the Examinee ID_C considers the test *Cidex*, which consist of question and answer, submitted for the exam. The event is inserted into the process of Examinee at the location where the test is sent to the manager.

- event AnswerDistributed(*idx*, *C_3*, ID_A) means that the Manager considers the encrypted answer *C_3*, associated with exaimer ID_A and masked examinee *idx* for marking. The event is inserted into the process of Manager at the location where the test is distributed to the examiner.

- event GiveMark(*idx*, *idex*, *answer*, *Mark*, ID_A) means that the examiner ID_A considers the test *idex*, which consists of *answer* evaluated with *Mark*. The event is inserted into the process of the process of examiner at the location where test is marked.

- event NotificationResult(IDc,idex,Mark) means that the Examinee ID_C accept the mark *Mark*. This event is inserted into the process of examinee at the location where mark is considered as accepted.

5.5.1 Model Choices

We model the public board as a public channel, and use the equational theory showed in Table 5.1. The theories consists of the standard equations for symmectric encryption, certificateless signcryption, digital signatures and session key generator. A sessions key in the sender is computed based on the a random value chosen by sender and public key of receiver, while session key in the receivers is computed based on the random public value sent by sender and his secret key correspondent to his previous public key. The senders session key and the receivers sessions key should same according the computational Diffie-Hellman's property.

Table 5.1: Equational theory to model OES Protocol

Primitive	Equation
`Symmectric Key Encryption`	$Dec(Enc(m, k), k) = m.$
`Certificateless Signatures`	$-getMess(sign(h1(m, id), l, ppk(r, msk, h0(id, EC(r), EC(x))), x)) = m.$ $-checksign(sign(h1(m, id), l, ppk(r, msk, h0(id, EC(r), EC(x))), x), EC(l), EC(ppk(r, msk, h0(id, EC(r), EC(x)))), EC(x)) = m.$
`Certificateless Signcryption`	$-SK1(l, EC(ppk(r, msk, h0(id, EC(r), EC(x)))), EC(x)) = SK2(EC(l), ppk(r, msk, h0(id, EC(r), EC(x))), x).$ $-clDec(clEnc(m, SK), SK) = m.$ $-checkcls(cls(h1(m, id), l, ppk(r, msk, h0(id, EC(r), EC(x))), x), EC(l), EC(ppk(r, msk, h0(id, EC(r), EC(x)))), EC(x)) = m.$ $-valuser(ppk(r, msk, h0(id, EC(r), EC(x))), EC(r), h0(id, EC(r), EC(x))) = true.$

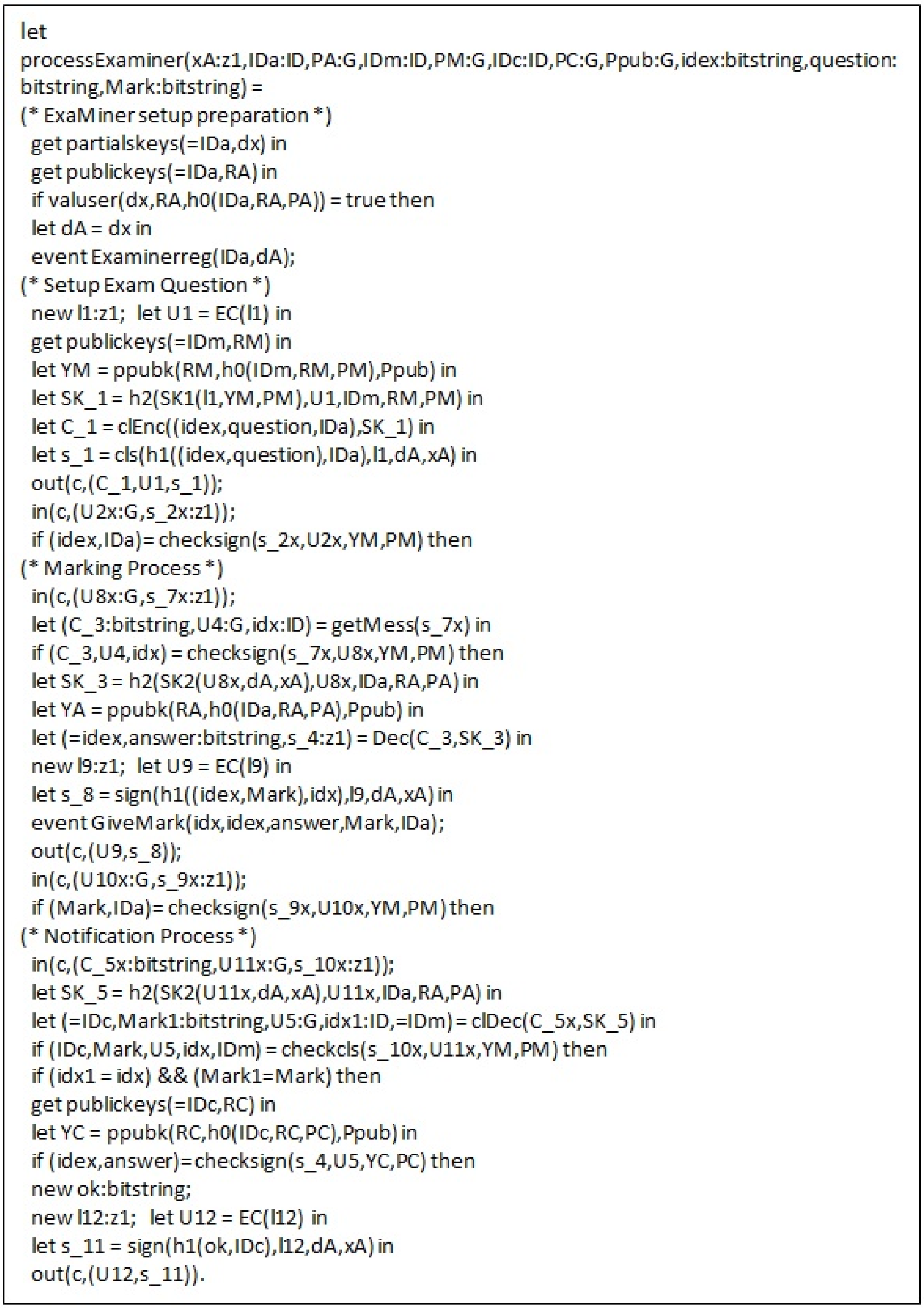
The process of the examiner, the examinee, the KGC (Key Generation Center), and the manager are shown in figure 5.1, figure 5.2, figure 5.3 and figure 5.4. While the exam process is depicted in figure 5.5.

5.5.2 Results

We use a formal verification program ProVerif to show the correct execution of the protocol. Assuming an attacker in control of the network and honest principals, ProVerif successfully proves all privacy and authentication requirements. Table 5.2 reports the execution of ProVerif. Also assuming corrupted principals, ProVerif proves the OES Protocol ensures all the requirements. Table 5.2 also reports the honest roles that are required for each requirement to hold. Note that we only model the processes needed to specify the requirement. For example, the specification of Anonymous Marking requires two honest participants, they are examinee and manager, otherwise, they could just reveal their test to the attacker, who would trivially violate the protocol. However, all other examinees can be corrupted and collude with the attacker to violate the protocol.

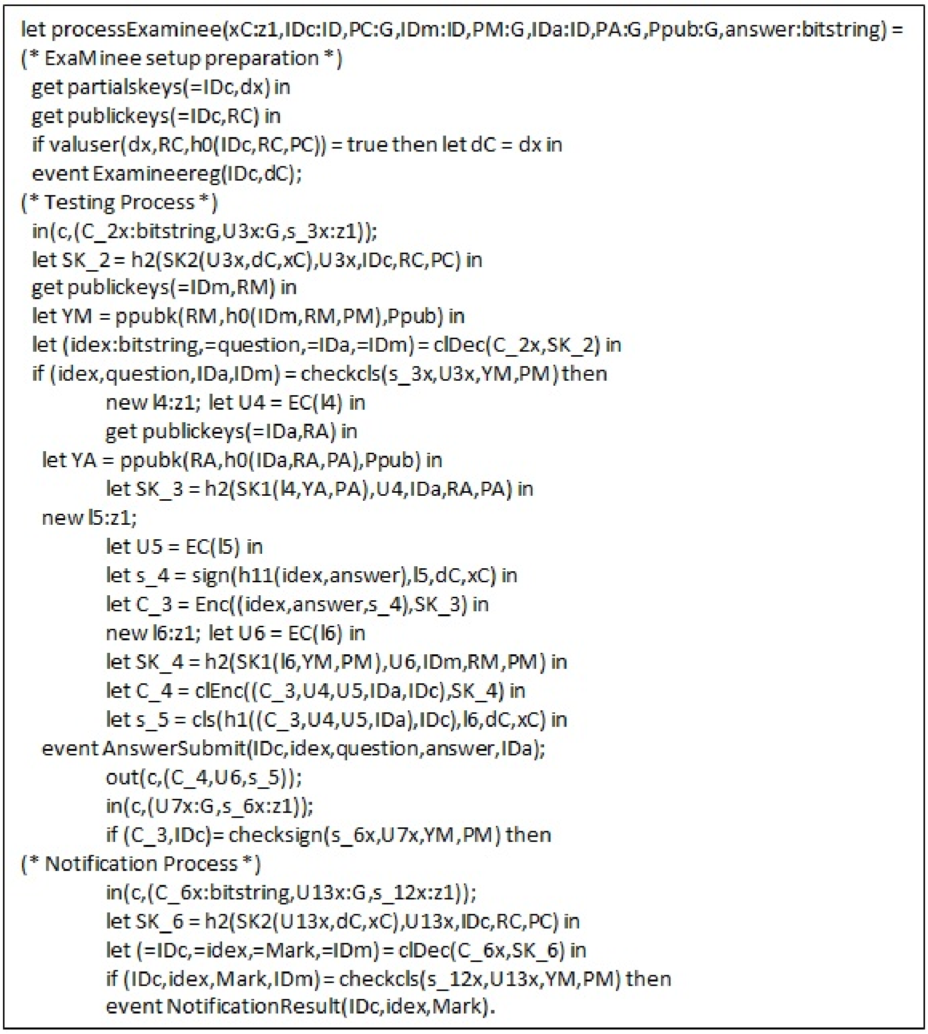
Table 5.2: Summary of privacy and authentication analysis of OES Protocol

Requirements	**Result**	**Honest Role**
`Question Confidentiality`	True	Examiner, Manager
`Answer Privacy`	True	Examiner, Examinee
`Mark Privacy`	True	Examiner, Manager
`Examiner authorization`	True	Examiner, Manager, KGC
`Examinee authorization`	True	Examinee, Manager, KGC
`Answer authenticity`	True	Examinee, Manager
`Test Origin authentication`	True	Manager
`Test authenticity`	True	Examiner, Manager
`Mark authenticity`	True	Examiner, Examinee, Manager
`Anonymous Marking`	True	Examinee, Manager



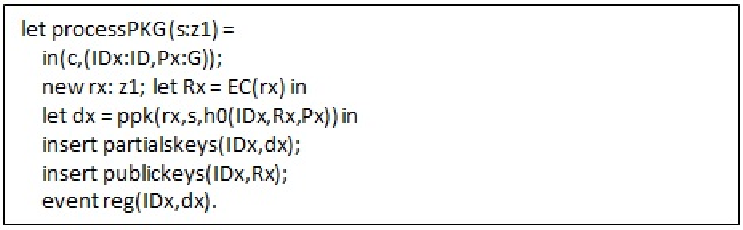
```
let
processExaminer(xA:z1,IDa:ID,PA:G,IDm:ID,PM:G,IDc:ID,PC:G,Ppub:G,idex:bitstring,question:
bitstring,Mark:bitstring) =
(* ExaMiner setup preparation *)
  get partialskeys(=IDa,dx) in
  get publickeys(=IDa,RA) in
  if valuser(dx,RA,h0(IDa,RA,PA)) = true then
  let dA = dx in
  event Examinerreg(IDa,dA);
(* Setup Exam Question *)
  new l1:z1;  let U1 = EC(l1) in
  get publickeys(=IDm,RM) in
  let YM = ppubk(RM,h0(IDm,RM,PM),Ppub) in
  let SK_1 = h2(SK1(l1,YM,PM),U1,IDm,RM,PM) in
  let C_1 = clEnc((idex,question,IDa),SK_1) in
  let s_1 = cls(h1((idex,question),IDa),l1,dA,xA) in
  out(c,(C_1,U1,s_1));
  in(c,(U2x:G,s_2x:z1));
  if (idex,IDa)= checksign(s_2x,U2x,YM,PM) then
(* Marking Process *)
  in(c,(U8x:G,s_7x:z1));
  let (C_3:bitstring,U4:G,idx:ID) = getMess(s_7x) in
  if (C_3,U4,idx) = checksign(s_7x,U8x,YM,PM) then
  let SK_3 = h2(SK2(U8x,dA,xA),U8x,IDa,RA,PA) in
  let YA = ppubk(RA,h0(IDa,RA,PA),Ppub) in
  let (=idex,answer:bitstring,s_4:z1) = Dec(C_3,SK_3) in
  new l9:z1;  let U9 = EC(l9) in
  let s_8 = sign(h1((idex,Mark),idx),l9,dA,xA) in
  event GiveMark(idx,idex,answer,Mark,IDa);
  out(c,(U9,s_8));
  in(c,(U10x:G,s_9x:z1));
  if (Mark,IDa)= checksign(s_9x,U10x,YM,PM) then
(* Notification Process *)
  in(c,(C_5x:bitstring,U11x:G,s_10x:z1));
  let SK_5 = h2(SK2(U11x,dA,xA),U11x,IDa,RA,PA) in
  let (=IDc,Mark1:bitstring,U5:G,idx1:ID,=IDm) = clDec(C_5x,SK_5) in
  if (IDc,Mark,U5,idx,IDm) = checkcls(s_10x,U11x,YM,PM) then
  if (idx1 = idx) && (Mark1=Mark) then
  get publickeys(=IDc,RC) in
  let YC = ppubk(RC,h0(IDc,RC,PC),Ppub) in
  if (idex,answer)=checksign(s_4,U5,YC,PC) then
  new ok:bitstring;
  new l12:z1;  let U12 = EC(l12) in
  let s_11 = sign(h1(ok,IDc),l12,dA,xA) in
  out(c,(U12,s_11)).
```

Figure 5.1: The process of Examiner



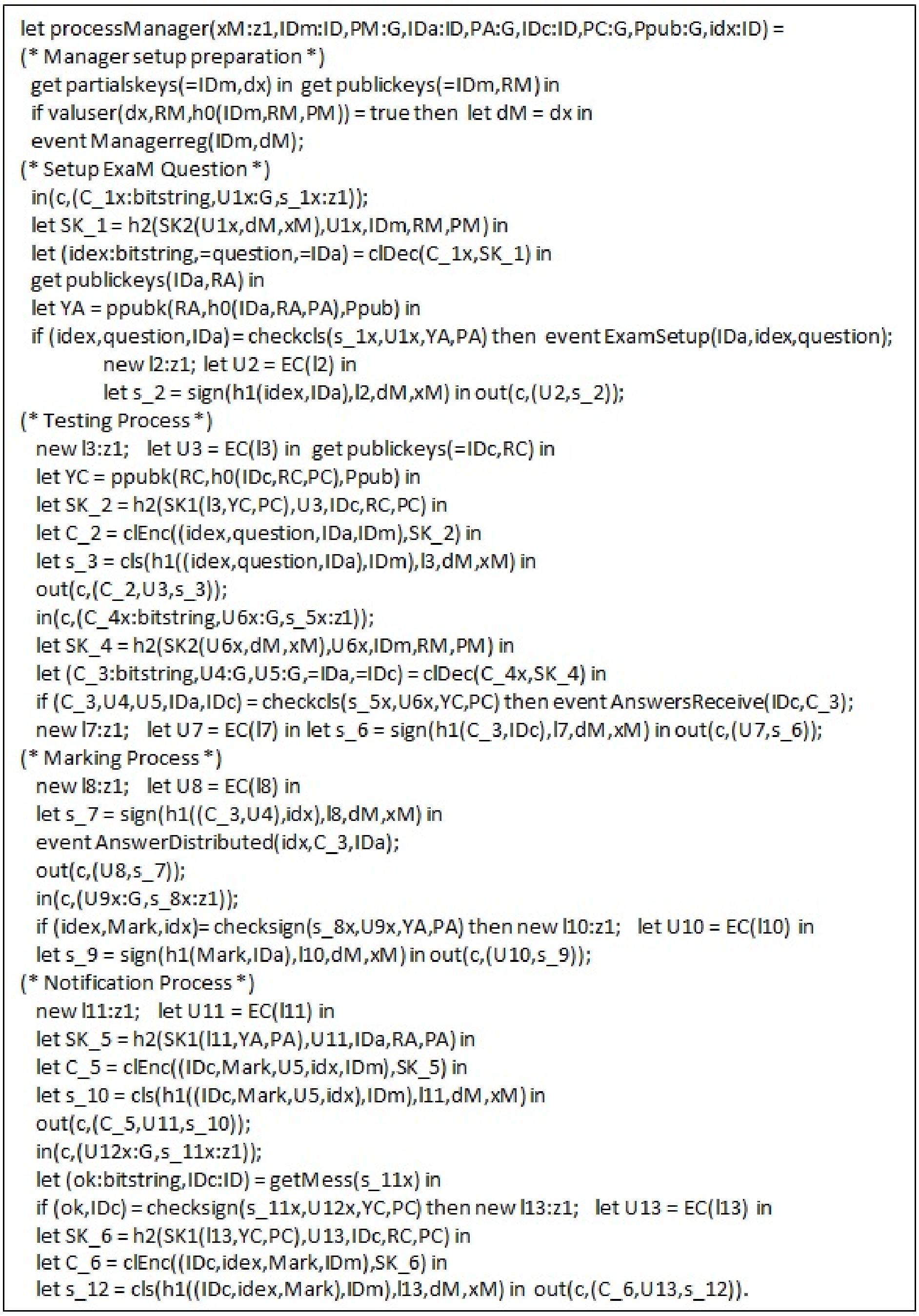
```
let processExaminee(xC:z1,IDc:ID,PC:G,IDm:ID,PM:G,IDa:ID,PA:G,Ppub:G,answer:bitstring) =
(* ExaMinee setup preparation *)
 get partialskeys(=IDc,dx) in
 get publickeys(=IDc,RC) in
 if valuser(dx,RC,h0(IDc,RC,PC)) = true then let dC = dx in
 event Examineereg(IDc,dC);
(* Testing Process *)
 in(c,(C_2x:bitstring,U3x:G,s_3x:z1));
 let SK_2 = h2(SK2(U3x,dC,xC),U3x,IDc,RC,PC) in
 get publickeys(=IDm,RM) in
 let YM = ppubk(RM,h0(IDm,RM,PM),Ppub) in
 let (idex:bitstring,=question,=IDa,=IDm) = clDec(C_2x,SK_2) in
 if (idex,question,IDa,IDm) = checkcls(s_3x,U3x,YM,PM) then
        new l4:z1;  let U4 = EC(l4) in
        get publickeys(=IDa,RA) in
  let YA = ppubk(RA,h0(IDa,RA,PA),Ppub) in
        let SK_3 = h2(SK1(l4,YA,PA),U4,IDa,RA,PA) in
  new l5:z1;
        let U5 = EC(l5) in
        let s_4 = sign(h11(idex,answer),l5,dC,xC) in
        let C_3 = Enc((idex,answer,s_4),SK_3) in
        new l6:z1;  let U6 = EC(l6) in
        let SK_4 = h2(SK1(l6,YM,PM),U6,IDm,RM,PM) in
        let C_4 = clEnc((C_3,U4,U5,IDa,IDc),SK_4) in
        let s_5 = cls(h1((C_3,U4,U5,IDa),IDc),l6,dC,xC) in
  event AnswerSubmit(IDc,idex,question,answer,IDa);
        out(c,(C_4,U6,s_5));
        in(c,(U7x:G,s_6x:z1));
        if (C_3,IDc)= checksign(s_6x,U7x,YM,PM) then
(* Notification Process *)
        in(c,(C_6x:bitstring,U13x:G,s_12x:z1));
        let SK_6 = h2(SK2(U13x,dC,xC),U13x,IDc,RC,PC) in
        let (=IDc,=idex,=Mark,=IDm) = clDec(C_6x,SK_6) in
        if (IDc,idex,Mark,IDm) = checkcls(s_12x,U13x,YM,PM) then
        event NotificationResult(IDc,idex,Mark).
```

Figure 5.2: The process of Examinee



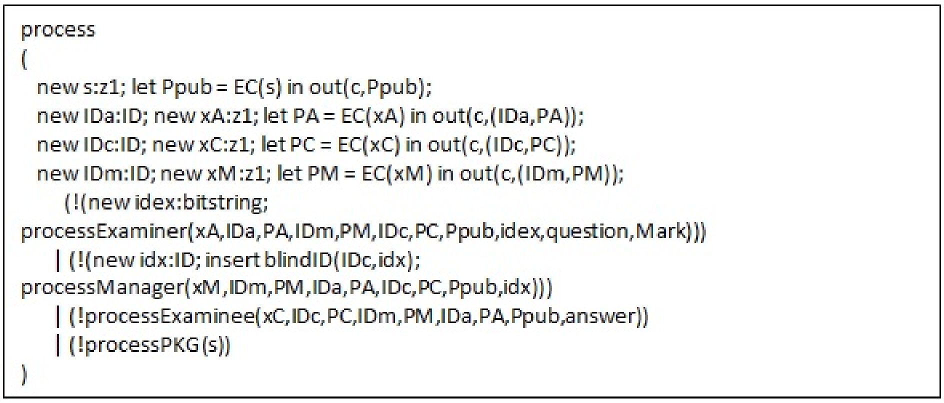
```
let processPKG(s:z1) =
  in(c,(IDx:ID,Px:G));
  new rx: z1; let Rx = EC(rx) in
  let dx = ppk(rx,s,h0(IDx,Rx,Px)) in
  insert partialskeys(IDx,dx);
  insert publickeys(IDx,Rx);
  event reg(IDx,dx).
```

Figure 5.3: The process of KGC



```
let processManager(xM:z1,IDm:ID,PM:G,IDa:ID,PA:G,IDc:ID,PC:G,Ppub:G,idx:ID) =
(* Manager setup preparation *)
 get partialskeys(=IDm,dx) in  get publickeys(=IDm,RM) in
 if valuser(dx,RM,h0(IDm,RM,PM)) = true then  let dM = dx in
 event Managerreg(IDm,dM);
(* Setup ExaM Question *)
 in(c,(C_1x:bitstring,U1x:G,s_1x:z1));
 let SK_1 = h2(SK2(U1x,dM,xM),U1x,IDm,RM,PM) in
 let (idex:bitstring,=question,=IDa) = clDec(C_1x,SK_1) in
 get publickeys(IDa,RA) in
 let YA = ppubk(RA,h0(IDa,RA,PA),Ppub) in
 if (idex,question,IDa) = checkcls(s_1x,U1x,YA,PA) then  event ExamSetup(IDa,idex,question);
          new l2:z1; let U2 = EC(l2) in
          let s_2 = sign(h1(idex,IDa),l2,dM,xM) in out(c,(U2,s_2));
(* Testing Process *)
  new l3:z1;   let U3 = EC(l3) in  get publickeys(=IDc,RC) in
  let YC = ppubk(RC,h0(IDc,RC,PC),Ppub) in
  let SK_2 = h2(SK1(l3,YC,PC),U3,IDc,RC,PC) in
  let C_2 = clEnc((idex,question,IDa,IDm),SK_2) in
  let s_3 = cls(h1((idex,question,IDa),IDm),l3,dM,xM) in
  out(c,(C_2,U3,s_3));
  in(c,(C_4x:bitstring,U6x:G,s_5x:z1));
  let SK_4 = h2(SK2(U6x,dM,xM),U6x,IDm,RM,PM) in
  let (C_3:bitstring,U4:G,U5:G,=IDa,=IDc) = clDec(C_4x,SK_4) in
  if (C_3,U4,U5,IDa,IDc) = checkcls(s_5x,U6x,YC,PC) then event AnswersReceive(IDc,C_3);
  new l7:z1;   let U7 = EC(l7) in let s_6 = sign(h1(C_3,IDc),l7,dM,xM) in out(c,(U7,s_6));
(* Marking Process *)
  new l8:z1;   let U8 = EC(l8) in
  let s_7 = sign(h1((C_3,U4),idx),l8,dM,xM) in
  event AnswerDistributed(idx,C_3,IDa);
  out(c,(U8,s_7));
  in(c,(U9x:G,s_8x:z1));
  if (idex,Mark,idx)= checksign(s_8x,U9x,YA,PA) then new l10:z1;   let U10 = EC(l10)  in
  let s_9 = sign(h1(Mark,IDa),l10,dM,xM) in out(c,(U10,s_9));
(* Notification Process *)
  new l11:z1;   let U11 = EC(l11) in
  let SK_5 = h2(SK1(l11,YA,PA),U11,IDa,RA,PA) in
  let C_5 = clEnc((IDc,Mark,U5,idx,IDm),SK_5) in
  let s_10 = cls(h1((IDc,Mark,U5,idx),IDm),l11,dM,xM) in
  out(c,(C_5,U11,s_10));
  in(c,(U12x:G,s_11x:z1));
  let (ok:bitstring,IDc:ID) = getMess(s_11x) in
  if (ok,IDc) = checksign(s_11x,U12x,YC,PC) then new l13:z1;   let U13 = EC(l13)  in
  let SK_6 = h2(SK1(l13,YC,PC),U13,IDc,RC,PC) in
  let C_6 = clEnc((IDc,idex,Mark,IDm),SK_6) in
  let s_12 = cls(h1((IDc,idex,Mark),IDm),l13,dM,xM) in  out(c,(C_6,U13,s_12)).
```

Figure 5.4: The process of Manager



```
process
(
  new s:z1; let Ppub = EC(s) in out(c,Ppub);
  new IDa:ID; new xA:z1; let PA = EC(xA) in out(c,(IDa,PA));
  new IDc:ID; new xC:z1; let PC = EC(xC) in out(c,(IDc,PC));
  new IDm:ID; new xM:z1; let PM = EC(xM) in out(c,(IDm,PM));
      (!(new idex:bitstring;
processExaminer(xA,IDa,PA,IDm,PM,IDc,PC,Ppub,idex,question,Mark)))
     | (!(new idx:ID; insert blindID(IDc,idx);
processManager(xM,IDm,PM,IDa,PA,IDc,PC,Ppub,idx)))
     | (!processExaminee(xC,IDc,PC,IDm,PM,IDa,PA,Ppub,answer))
     | (!processPKG(s))
)
```

Figure 5.5: The exam process

Chapter 6

Concluding Remarks

6.1 Conclusion

In this study of online examination system, we manage to overcome some security issues of the system and some cheating issues from all parties by establishing a basic framework. This framework combines the online examination web application, network system configuration, and communication protocol as an integrated system.

In the context of web application, we combine several techniques for cheating prevention like Fisher-Yates random question, time limit, automatic scheduling, and seating arrangement.

In the context of network configuration, we use the combination of firewall in the server, proxy and MMC in the client system as a security guarantee for the online examination systems, both against cheating attempts by examinees and attacks by third party.The firewall in a server functions as a blocker of access attempts to the web server from outside, except accessing from the proxy that has been registered on that firewall and can function as scanning port guard. Proxy server installed on the client functions to block all access to the outside by examinees, except accessing to the online examination server. MMC setting in the client functions well as a blocker of examinee's attempts to access external drives on the PC that they used for the online examination, as well as to block the use of Windows application programs that can be used by the examinee to cheat, such as Windows messenger, Windows mail, and so on.

We have designed and implemented certificateless signcryption, CLSC, based on the elliptic curve. Our scheme meets all of the basic security needs such as message authentication, integrity, unforgeability, non-repudiation and forward secrecy. This scheme is implemented using javascript and utilizes several existing libraries such as cryptoJS, sec_mod.js from spruegel and ec.js owned by Tom Wu. In the scalar multiplication operation with the point on the elliptic curve, we have used the windowed method approach which is faster than the double-and-add approach because it uses less point summation (which in practice is slower than doubling). Our scheme is more efficient than the previous schemes because our CLSC based on elliptic curve which is more efficient than bilinear pairings and finite field exponentiations used in the previous CLSC schemes. Besides, our scheme offers shorter ciphertext size than previous CLSC schemes.

At last, we have constructed an OES protocol scheme based on certificateless signcryption. We have shown how to model an OES protocol in the applied phi calculus, and defined ten relevant security properties, four privacy property and six authentication properties. We have analyzed the security of our OES protocol scheme using ProVerif software. ProVerif shows that our scheme is secure under the formal model analysis.

6.2 Future Work

As a future work, we plan to analyze our OES protocol under computational model to ensure our security properties. We can use some software under computational approach such as CyptoVerif which uses applied phi-calculus to model their process.

Publications

1. Abdul Wahid, Yasushi Sengoku, Masahiro Mambo, "Toward Constructing a Secure Online Examination System", *Proceedings of the 9^{th} International Conference on Ubiquitous Information Management and Communication, ACM IMCOM 2015, Article No. 95, January 2015. 8 pages.*

2. Abdul Wahid and Masahiro Mambo,"Implementation of Certificateless Signcryption Based on Elliptic Curve using Javascript" in *IJCANDI (International Journal of Computing and Informatics) Vol.1 No.3, pp. 90-100, August 2016.*

Bibliography

[1] M. Sarrayrih and M. Ilyas, "Challenges of Online Exam , Performances and problems for Online University Exam", Int. J. Comput. Sci., vol. 10, no. 1, pp.439-443, 2013.

[2] P. Guo, H. F. Yu, and Q. Yao, "The research and application of online examination and monitoring system", Proc. 2008 IEEE Int. Symp. IT Med. Educ. ITME 2008, pp.497-502, 2008.

[3] O. T. Oluwatosin and D. D. Samson, "Computer-Based Test : Security and Result Integrity", Int. J. Comput. Inf. Technol. (ISSN 2279 - 0764), vol. 02, no. 02, pp.32-329, 2013.

[4] S. William, Cryptography and Network Security, vol. 139, no. 3. Boston, USA.: Prentice Hall, 2011.

[5] V. S. Miller, "Short Programs for functions on Curves", Unpublished, pp. 17, 1986.

[6] K. Aqeel, S. Kuldip, and S. Sandeep, "Implementation of elliptic curve digital signature algorithm", Int. J. Comput. Appl., vol. 2, no. 2, pp. 2127, 2010.

[7] R.-J. Hwang, C.-H. Lai, and F.-F. Su, "An efficient signcryption scheme with forward secrecy based on elliptic curve", Appl. Math. Comput., vol. 167, pp. 870881, 2005.

[8] D. Denning and G. Sacco, "Timestamps in key distributed protocols", Commun. ACM, vol. 24, no. 8, pp. 533535, 1981.

[9] B. Blanchet, B. Smyth, and V. Cheval, "ProVerif 1.88: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial", 2013.

[10] B. Blanchet, "CryptoVerif Computationally Sound , Automatic Cryptographic Protocol Verifier User Manual", pp. 141, 2011.

[11] G. R. Cluskey, C. R. Ehlen, and M. H. Raiborn, "Thwarting online exam cheating without proctor supervision", J. Acad. Bus. Ethics, vol. 4, pp.1-8, 2011.

[12] J. Castella-Roca, J. Herrera-Joancomarti, and A. Dorca-Josa, "A secure e-exam management system", First Int. Conf. Availability, Reliab. Secur. ARES 2006, pp.864-871, 2006.

[13] B. P. U. Ivy, "WebBased online Secured Exam", Int. J. Eng. Res. Appl., vol. 2, no. 1, pp. 943944, 2012.

[14] S. Liu and Q. Gong, "The research on anti-cheating strategy of online examination system", 2011 2nd Int. Conf. Artif. Intell. Manag. Sci. Electron. Commer. AIMSEC 2011 - Proc., pp. 17381741, 2011.

[15] N. Chiranji, C. Depthi, and T. P. Shekhar, "A Novel Approach to Enhance Security for Online Exams", Int. J. Comput. Sci. Technol., vol. 2, no. 3, pp. 8489, 2011.

[16] B. Hang, "The Design and Implementation of On-Line Examination System", 2011 Int. Symp. Comput. Sci. Soc., no. 1, pp. 227230, 2011.

[17] Z. Islam, M. Rahman, and K. Islam, "Online examination system in bangladesh context", Sci. Environ. Technol. , vol. 2, no. 3, pp. 351359, 2013.

[18] A. Wahid, Y. Sengoku, and M. Mambo, "Toward constructing a secure online examination system", in Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication - IMCOM 15, 2015, pp. 18.

[19] A. OluAde-Ibijola, "A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-specific Data Structures", Int. J. Comput. Appl., vol. 54, no. 11, pp. 2428, 2012.

[20] E. Taderera, L. Nyikahadzoi, W. Matamande, and E. Mandimika, "Exploring management strategies to reduce cheating in written examinations: case study of Midlands State University", J. Case Stud. Educ., vol. 5, pp. 113, 2014.

[21] S. Sabri, "Item Analysis of Student Comprehensive Test for Research in Teaching Beginner String Ensemble Using Model Based Teaching Among Music Students in Public Universities", Int. J. Educ. Res., vol. 1, no. 12, pp. 114, 2013.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes", in Advances in Cryptology-CRYPTO84, 1984, pp. 4753.

[23] T. Okamoto, "A Single Public-Key Authentication Scheme for Multiple Users", IEICE Trans., vol. J69-D, no. 10, pp. 14811489, 1986.

[24] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", SIAM J. Comput., vol. 32, no. 3, pp. 586615, 2003.

[25] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography", Adv. Cryptology-ASIACRYPT 2003, pp. 140, 2003.

[26] Y.Zheng, "Digital signcryption or how to achieve cost (signature and encryption) cost (signature)+cost (encryption)", Adv. Cryptol. Crypto '97,March, pp. 165179, 1997.

[27] D. Hankerson, A. J. Menezes, and S. Vanstone, Guide to Elliptic Curve Cryptography. Springer-Verlag New York Inc, 2006.

[28] S. Seo and E. Bertino, "Elliptic Curve Cryptography based Certificateless Hybrid Signcryption Scheme without Pairing", CERIAS Tech Rep. 2013-10, 2013.

[29] J. Won, S.-H. Seo, and E. Bertino, "A Secure Communication Protocol for Drones and Smart Objects", in ASIA CCS15, 2015, pp. 249260.

[30] M. Barbosa and P. Farshim, "Certificateless Signcryption", ACM Symp. Information, Comput. Commun. Secur., pp. 369372, 2008.

[31] X. Zheng and X. Yang, "Improvement of a Certificate less Signcryption Scheme without pairing", Int. J. Sci., vol. 2, no. 7, pp. 8187, 2015.

[32] S. Wenbo, K. Neeraj, G. Peng, and Z. Zezhong, "Cryptanalysis and improvement of a certificateless signcryption scheme without bilinear pairing", Front. Comput. Sci., vol. 8, no. 4, pp. 656666, 2014.

[33] J. Baek, R. Steinfeld, and Y. Zheng, "Formal proofs for the security of signcryption", J. Cryptol., vol. 20, no. 2, pp. 203235, 2007.

[34] L. Chen and J. Malone-Lee, "Improved Identity-Based Signcryption", Public Key Cryptogr. 2005, vol. 3386, pp. 362379, 2005.

[35] R. Steinfeld and Y. Zheng, "A Signcryption Scheme Based on Integer Factorization", Inf. Secur. LNCS, vol. 1975, pp. 308322, 2000.

[36] Y. Zheng and H. Imai, "How to construct efficient signcryption schemes on elliptic curves", Inf. Process. Lett., vol. 68, no. 5, pp. 227233, 1998.

[37] Z. Jin, Q. Wen, and H. Du, "An improved semantically-secure identity-based signcryption scheme in the standard model", Comput. Electr. Eng., vol. 36, no. 3, pp. 545552, 2010.

[38] Y. Yu, B. Yang, Y. Sun, and S. lin Zhu, "Identity based signcryption scheme without random oracles", Comput. Stand. Interfaces, vol. 31, no. 1, pp. 5662, 2009.

[39] A. Huszti and A. Petho, "A secure electronic exam system", Publ. Math. Debrecen, vol. 3, no. 4, pp. 209312, 2010.

[40] J. Dreier, R. Giustolisi, A. Kassem, P. Lafourcade, G. Lenzini, and P. Y. A. Ryan, "Formal Analysis of Electronic Exams", Proc. 11th Int. Conf. Secur. Cryptogr., 2014.

[41] R. Giustolisi, G. Lenzini, and P. Y. A. Ryan, "Remark!: A Secure Protocol for Remote Exams", Secur. Protoc. XXII, LNCS. Springer, pp. 3848, 2014.

[42] The PHP Group, "Mcrypt Function, Manual book", `http://php.net/manual/en/book.mcrypt.php`, accessed Jan. 20. 2015.

[43] T.M. Eastep, "Shorewall 4.4/4.5/4.6 Documentation, technical Documentation", `http://shorewall.net/Documentation_Index.html`, accessed Dec. 12. 2014.

[44] F. Chemolli, "Squid-cache Wiki, Online Manuals", 2000. `http://wiki.squid-cache.org/ConfigExamples#Online_Manuals`, accessed Oct. 10. 2014.

[45] Microsoft Developer, "Step-by-step Guide to the Microsoft Management Console, Technical Documentation", `http://msdn.microsoft.com/en-us/library/bb742442.aspx`, accessed Jan. 20. 2015.