PRIVACY-PRESERVING BASED AUTOMATED TRUST NEGOTIATION IN E-LEARNING SYSTEM

MALANH PHETSAVONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTING IN ENGINEERING SYSTEMS

INTERNATIONAL COLLEGE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2015

KMITL-2014-IC-M-011-004

PRIVACY-PRESERVING BASED AUTOMATED TRUST NEGOTIATION IN E-LEARNING SYSTEM

MALANH PHETSAVONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTING IN ENGINEERING SYSTEMS

INTERNATIONAL COLLEGE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2015

KMITL-2014-IC-M-011-004

THESIS TITLE	PRIVACY-PRESERVING BASED AUTOMATED TRUST NEGOTIATION IN E-LEARNING SYSTEM
STUDENT NAME	MR. MALANH PHETSAVONG
STUDENT ID	56610017
DEGREE	MASTER OF ENGINEERING
PROGRAM	COMPUTING IN ENGINEERING SYSTEMS
ADVISOR	DR. PIKULKAEW TANGTISANON

ABSTRACT

Currently, there are many kinds of the internet-based services, such as social networks, file sharing and others. The numbers of the services are rapidly increasing. Many services expect to establish mutual trust among the users. Unfortunately, there is no standard method on the internet to establish trust among the user; different entities use different methods to learn whether the user is trusted or not. The process of exchanging credentials between two negotiators, through a sequence of alternating credentials request and disclose, called trust negotiations. An approach to determine the automatic exchange of sensitive credentials between people, without previous trust relationship, by using access control policies is called Automated Trust Negotiation (ATN). This thesis examines secret exchanging on the examination system to protect both a student's privacy and teacher's sensitive information. In the beginning, the student and the teacher exchange their credentials with privacy scheme for authentication. At the end of the protocol, in the score computation process, the teacher learns nothing from the students and the students learn only scores they get not the answer keys.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Pikulkaew Tangtisanon for her steady support during my master studies. She has provided me a lot of support and freedom that I needed to be successful in my research. Her insights and suggestions have been invaluable to my research.

I wish to express acknowledgement to AUN/SEED-Net for awarding me the scholarship with the financial support for master degree within 2 years, and the research Fund of King Mongkut's Institute of Technology Ladkrabang funding under the contract KREF125609. Moreover, I extend my sincere appreciation to King Mongkut's Institute of Technology Ladkrabang (KMITL) for giving me the great opportunity to do research in warmly and friendly environment.

I gratefully acknowledge goes also to all of professor, lectures and supporting staffs in International College, who always to help and give me guidelines and conveniences during the whole period of my master study.

Finally, I would like to acknowledge my family for their unconditional support, especially my mother for support me while I returned to graduate school to obtain this master degree.

Malanh Phetsavong

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

The development of the internet technology has been rapidly improved so anyone can get access to the Internet from anywhere in anytime. There are many new services in the internet. E-learning is one of the most popular applications. Many universities offer online course where students learn, take examination and get a degree online. There are tremendous of research focusing on e-learning framework to improve learning system with enhancement of new technology while the needs of the user privacy have been ignored.

1.2 Motivation

Nowadays, communication is one of the main method of people life. Some of the services need to disclose their information such as name, birthday and so on, some of each information are sensitive that they do not want to disclose to each other such as address, telephone number. Hence, privacy preserving is more important for people to prevent their sensitive information. For example, suppose Alice want to buy a beer in Beer store, Beer store allow Alice to buy a beer if she over 18 years old, and Alice has a digital license that include Alice's birthday and address, Alice has to disclose her digital driver license to Beer store even she does not want to disclose her address to Beer store. Another example, for score computation in e-learning system, in the case that the pre-test and post-test contain the same set of questions to measure the improvement of the student. The celebrity student may not want to reveal his result of the pre-test to anyone including the teacher while the teacher is not willing to share answer key to the student so she must use this test again at the end of the course for the post-test.

1.3 Objective

- To simulate eager strategy and parsimonious strategy.

- To apply Privacy Preserving Based Automated Trust Negotiation into E-learning system in order to protect privacy of a teacher and a student.

1.4 Thesis Organization

This thesis consists of five main chapters, given briefly summarized as follows.

Chapter 2 describes the trust negotiation fundamental and trust strategies, and review the score computation in subjective test system. Some e-learning system has also introduced in this chapter. The cryptographic tools which is used to prevent score computation data, have used in this chapter.

Chapter 3 presents the proposed methods and tools that support in this thesis.

Chapter 4 gradually shows the results.

Chapter 5 is the last chapter that provides conclusions and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Automated Trust negotiation

2.1.1 Overview

Blaze, Feigenbaum, and Lacy [1] introduced the term trust management to group together some principles dealing with decentralized authorization. In trust management [1-5], access control decisions are based on authenticated attributes of the subjects, which are established by digitally signed credentials. Each credential associates a public key with the key holder's identity and/or attributes such as employer, group membership, credit card information, birth-date, citizenship, and so on. Because these credentials are digitally signed, they can serve to introduce strangers to one another without online contact with the attribute authorities.

Winsborough, Seamons, and Jones [6] introduced the notion of *automated trust negotiation* (ATN). The goal of ATN [6-10] to enable the trust establishment between negotiators in an open environment, such as the Internet. Access Control Policies (ACP) are established to determine the credentials disclosure then grant them system resources. One of the negotiation methodologies is done through a sequence of exchanges credentials and rules that begin by disclosing non sensitive credentials. As more credentials and the access control policies are disclosed, higher levels of mutual trust are established. Finally, when the access control policy of the service is satisfied, the mutual trust is successfully established. Trust negotiation differs from trust management in that:

- In trust negotiation, credentials are modeled as sensitive information, and protected by access control policies just same as other resources in the system.
- Trust negotiation is performed in a peer-to-peer architecture, where a client and a server are treated equally. Instead of a one-shot authorization, trust is established incrementally through a sequence of credential disclosure.

In general, Trust negotiation is proposed and emphasized in access control area for resource sharing and cooperation in open systems. Since nowadays applications are mostly

service oriented and cross-boundary, entities involved in the access control process are usually unfamiliar, so traditional access control mechanism, which basically uses the identity of the involved entities to control authorization internal the organization, is no longer sufficient. Then, attributes and policies are introduced to describe entity characteristic and security requirement respectively [11], and trust negotiation is developed to help the involved entities exchanging attribute and policy. So the trust degree between the unfamiliar entities can be improved.

The most concerned issues in trust negotiation include:

- Protection to sensitive information [12]. Because the entities are unfamiliar with each other, they are unwilling to provide their information all to someone they don't trust, especially in military systems. However, information sharing is inevitable in open systems, so particular mechanism should be adopted to restrict access to sensitive information.
- Efficiency of trust negotiation. The purpose of trust negotiation is to help collaborating entities establishing mutual trust before they get into collaboration. If the cost of negotiation is huge, even bigger than the cost of collaboration, then the collaboration itself would seems to be meaningless.
- Success rate of negotiation. Together with security, successful collaboration is desired to every entity involved in the access process. If all of them define rigorous policy to protect benefit of themselves, and don't make any concession, the compromise would be difficult to achieve. So a good trust negotiation mechanism should adopt approaches to increase success rate.

These issues are interrelated with each other. Nowadays products of automated trust negotiation (ATN) [6] are always concentrating on one or two of the issues, and can hardly do well in all of them.

In traditional ATN approaches the only way to use a credential is to send it as a whole, thus disclosing all the information in the credential. In other words, a digital credential is viewed as a black-box, and the information in a credential is disclosed in an all-or-nothing fashion. In these approaches sensitive attribute values stored in a credential are protected using access control techniques. There is an access control policy associated with each credential and

a credential can be disclosed if its access control policy has been satisfied. Viewing a credential as a black-box severely limits the power of ATN. The following are some of the limitations.

- Because attribute information is disclosed in an all-or-nothing fashion, each attribute can be disclosed only when the policy governing the credential and its entire contents is satisfied, leading to unnecessary failure. For example, suppose Bob would allow Alice to access a resource provided Alice is over 21, and Alice has a digital driver license that includes Alice's birth-date and address. If Alice does not want to reveal her address (or her exact birth-date) to Bob, the negotiation would fail, even if Alice were willing to prove she is over 21.
- When one negotiator does not want to disclose detailed information about his policy and the other negotiator does not want to disclose too much information about her attributes, a negotiation can fail even though the amount of information that needs to be disclosed by each party is acceptable to both. For example, suppose Bob is a bank that offers a special-rate loan and Alice would like to know whether she is eligible for such a loan before she applies. Bob is willing to reveal that his loan approval policy uses one's birth-date, current salary, and the length of the current employment; however, Bob considers further details of this policy to be a trade secret that he is unwilling to reveal. Alice would like to know whether she is eligible for the loan while disclosing as little information about her attributes as possible. In particular, Alice does not want to disclose the exact values of her birth-date or salary level. Using traditional ATN techniques, this negotiation would fail.
- If there is a cyclic dependency among credentials and their policies, negotiations can fail unnecessarily. For example, in a negotiation between Alice and Bob, suppose Alice has a credential c_1 that can be disclosed only if Bob has c_2, and Bob has c_2, but can disclose it only if Alice has c_1. Using traditional ATN techniques, the negotiation would fail because neither c_1 nor c_2 can be disclosed before the other, even though allowing Alice and Bob to exchange both c_1 and c_2 would not violate either negotiator's policy.

In trust negotiation [6-7,9], the disclosure of a credentials s is controlled by an access control policy p_s that specifies the prerequisite condition that must be satisfied in order for credential s to be disclosed. Typically, the prerequisite condition are a set of credentials $C \subseteq$

c, where c is the set of all credentials. As in [6], the policies in this chapter are modeled using propositional formulas. Each policy p_s takes the from $s \leftarrow \phi_s(c_1, \dots, c_k)$ where $c_1, \dots, c_k \epsilon c$ and $\phi_s(c_1, \dots, c_k)$ is a normal formula consisting of literals c_i, the Boolean operators $\wedge$ and $\vee$. In this chapter, s is referred to as the target of p_s, and $\phi_s(c_1, \dots, c_k)$ is referred to as the policy function of p_s.

Given a set of credentials $C' \subseteq C$ and a policy function $\phi_s(c_1, \dots, c_k)$. We denote $\phi_s(C')$ as the value of the normal formula $\phi_s(x_1, \dots, x_k)$ where $x_i = 1$ if and only if $\phi_s(C') = 1$. During trust negotiation, one can disclose credential s if $\phi_s(C') = 1$ where $C'$is the set of credentials that he or she has received from the other party.

A trust negotiation protocol is normally initiated by a client requesting a resource from a server. The negotiation consists of a sequence of credentials exchange Trust is established if the initially requested resource is granted and all policies for disclosed credentials are satisfied [6]. In this case, the negotiation between the client and server is a *successful* negotiation, and otherwise, it is a *failed* negotiation. The formal definition for traditional trust negotiation as follows:

Definition 2.1 (Traditional Trust Negotiation): Let C_s and P_s ($C_c$and P_c) be the sets of credentials and policies possessed by a negotiating server (client). The negotiation is initiated by a request for $s \in C_s^1$ from the client. The goal of trust negotiation is to find a credential disclosure sequence $(c_1, \dots, c_n) = s$, where $c_i \in C_s \cup C_c$, and such that for each c_i, $1 \leq i \leq n$, the policy for c_i is satisfied by the credentials already disclosed, i.e., $\phi_{C_i}\left(\cup_{j<i} c_j\right) = 1$. If the client and server find a credential disclosure sequence, the negotiation succeeds, otherwise, it fails.

client ($n_c = 4$)		server ($n_s = 4$)	
q_1 :	$c_1 \leftarrow s_1 \wedge s_2$	p_1 :	$R \leftarrow c_1 \vee c_2$
q_2 :	c_2	p_2 :	s_1
q_3 :	c_3	p_3 :	s_2
q_4 :	c_4		

Figure 2.1: Example of trust policies

Figure 2.1 shows an example of policies owned by the client and the server. Credentials owned by the server and the client are represent by s_1, s_2 and c_1, c_2 respectively, the number of policies is represent by $n_c$and n_s, where R is represent for a target of service.

2.1.2 Trust Target Graph Protocol

In this protocol, a trust negotiation process involves the two negotiators working together to construct a trust target graph (TTG). A TTG is a directed graph. Each node is either a trust target or a linking goal. When a requester requests access to a resource, the access mediator and the requester enter into a negotiation process. The access mediator creates a TTG containing one target, which call the primary target. The access mediator then tries to process the primary target, and sends the partially processed TTG to the requester. In each following round, one negotiator receives from the other new information about changes to the TTG, verifies that the changes are legal, and updates its local copy of the TTG accordingly. The negotiator then tries to process some nodes, making its own changes to the graph, which it then sends to the other party, completing the round. The negotiation succeeds when the primary target is satisfied; it fails when the primary target is failed, or when a round occurs in which neither negotiator changes the graph.

Figure 2.2 shows the logical relationship between the client and the server that represented in a single trust target graph. There are eight rules in this negotiation process which are $s_1 \wedge s_2, c_2, c_3, c_4, c_1 \vee c_2, s_1$, and s_2 .

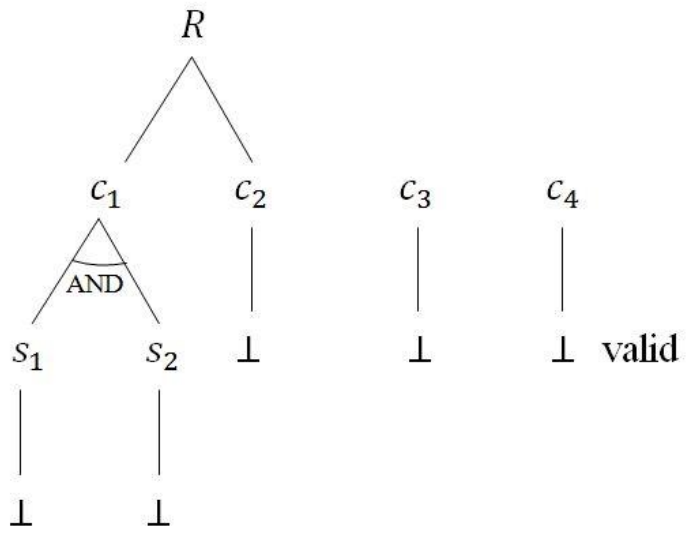

Figure 2.2: Example of trust target graph

2.1.3 Negotiation Strategies

Trust negotiations strategies govern the exchange of credentials between participants, in the similar sense that a network protocol governs the exchange of data between systems. In a trust negotiation framework, the negotiation strategy controls the search for successful negotiation. The basic principle behind the negotiation process is the iterative bilateral disclosure of credentials, in which each disclosure "unlocks" the other parties' credentials which can then be disclosed.

Thus communication consists of credentials requests, credentials counter requests and credential disclosure. For that reason the negotiation process or cycle has to begin with the exchange of non-sensitive credentials. With rise of mutual trust, more and more sensitive credentials that satisfy policies are exchanged, enabling further credential exchange and ending with adequate set of credentials needed.

This is similar to real life analogy where people gradually, thought, communication learn about each other and earn each other's trust.

Since the exchange has to begin with non-sensitive or unprotected credentials it's obvious that if all credentials are protected, establishing trust is not possible.

Depending on the dependencies and the number of credentials needed the process can end within a single round or it may never end, if dependencies prove to be cyclic, forming a deadlock or one side doesn't have the required credentials.

Ideally a strategy should lead to successful negotiation whenever possible and terminate with failure when impossible. It should also do so without disclosing any non-essential credentials.

Additionally to time complexity, communication request tend to grow in size increasing space complexity adding to overall communication complexity.

Each negotiation needs to have five basic properties:

1. **Safety**, i.e. credentials are not disclosed until the ACP is satisfied,
2. It should lead to a successful negotiation whenever possible, i.e. it should be **complete**
3. It should **terminate** with failure when success is impossible
4. It should use **minimal** set of credentials during negotiation
5. It should be reasonably **efficient** in the sense of communication and computational cost.

Regardless of the strategy in use, success is not always possible. Either side may not have the necessary credentials, or the ACPs protecting the credentials may form a cyclic dependence which can't be resolved. Because of such scenarios strategies must have a failsafe mechanism in the form of a timeout counter, i.e. if mutual trust is not established within *n* iterations the process terminates with failure.

According to [6] there are two distinct basic trust negotiation strategies: eager and parsimonious

2.1.3.1 Eager Strategy

The negotiators take a turn to exchange every currently unlocked credentials to each other. As credentials are exchanged, more credentials are able to unlock. The negotiation is terminated when a negotiator receives the same set of credentials from the other. The negotiators make little or no use of credential request.

Definition (Eager Negotiation): A trust negotiation, $\{C_i\}_{i\in[0,m]}$, is an eager negotiation if,

1. C_0 is the maximal set such that $unlocked(C_0, \emptyset)$ and, for all a $i \in [1, m]$, C_i is the maximal set that $unlocked(C_i, C_{i-1})$, and,
2. For all $i \in [1, m-3]$, $C_i \neq C_{i+2}$, and,
3. If m is even, $C_{m-2} \neq C_m$.

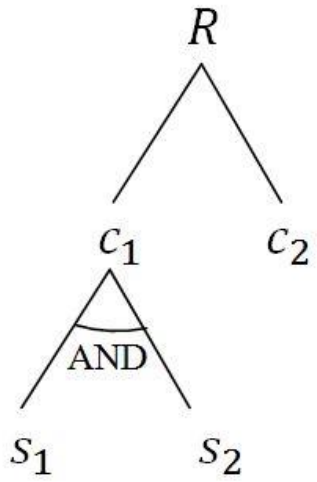

Figure 2.3: Example of Eager strategy

2.1.3.2 Parsimonious Strategy

Eager negotiation is an approach to disclose all of the credentials that can be disclosed to each other. In parsimonious strategy, each negotiator begins by repeats sending request for credentials to each other. An intuitive explanation of Parsimonious strategy as below;

1. Requests are exchanged to guide the negotiation toward satisfying a particular trust target, ψ. In general, this trust target could be a SGP or a trust requirement set by the client. To simplify the presentation, the trust target is assumed to be a SGP. The specification and ensuing results can easily be generalized to cover the case of a trust target set by the client as well. Under this assumption, the first credential request from the server, ψ_1, is the trust target (i.e., the SGP). (The content of ψ_0 is irrelevant and undefined.)

2. When and if a request is sent that can be satisfied by unprotected (and therefore unlocked) credentials, the negotiation reaches the point of confidence, the negotiation is bound to succeed.
3. Initial credential disclosures are empty in each stage up to and including the point of confidence. If the point of confidence is never reached, the negotiation terminates without disclosing any credentials.
4. Prior to the point of confidence, each successive credential request is derived from its predecessor in a manner that makes satisfying that request a necessary and sufficient condition for a disclosure to unlock credentials that satisfy the predecessor.
5. After the point of confidence is reached, the client resends its prior requests, going through them backwards, at the same time disclosing appropriate credentials to unlock solutions to those requests.
6. As mentioned above in point 2, when and if a request is sent that can be satisfied by a set of unprotected credentials, a minimal such set is disclosed in the next stage. Each successive step also discloses a minimal credential set that satisfies a credential request, working backwards through the requests that were issued prior to reaching the point of confidence. The client, which drives the negotiation, will have recorded each of the requests that has flowed. It refers to requests it received from the server when selecting its own credential disclosures; it resends the requests it sent to the server, as outlined in point 5 above. Each disclosure unlocks the next, until a disclosure satisfying the original trust target is unlocked.

Definition (Parsimonious Negotiation): Let target credential expression be ψ. To be a parsimonious negotiation with respect to the trust target, ψ, a trust negotiation, $\{C_i\}_{i\in[0,m]}$, must be accompanied by a sequence of credential request, $\{\psi_i\}_{i\in[0,m]}$, and must satisfy the following six requirements:

1. $\psi = \psi_1$
2. If there exists a $j \in [1, m]$ and a $C \subseteq AltCreds_{j+1}$ such that $sat(C, \psi_j)$ and $unlocked(C, \emptyset)$, then letting k be the least such j, the negotiation reaches the point of confidence at stage k. Otherwise, let $k = m$
3. For all i, $1 \leq i \leq k, C_i = \emptyset$
4. For all i, $1 \leq i \leq k, \psi_i$ and ψ_{i+1} have the following relationship:

a. For all $C \subseteq AltCreds_{i+2}$, $sat(C, \psi_{i+1})$ if and only if there exists a $C' \subseteq AltCreds_{i+1}$, such that $sat(C', \psi_i)$ and $unlocked(C', C)$,

5. After reaching the point of confidence, prior client request (which have even indices) are replayed. If k is even, require $\psi_{k+j} = \psi_{k-j}$ for even j, $2 \leq j \leq m - k$ (if any). If k is odd, require $\psi_{k+j} = \psi_{k-j}$ for odd j, $1 \leq j \leq m - k$ (if any).
6. If the negotiation reaches the point of confidence at stage k, require that for all j, $0 \leq j \leq m - k$ (f any) C_{k+j+1} is a minimal subset of $AltCreds_{k+j+1}$ satisfying $sat(C_{k+j+1}, \psi_{k-j})$ (and, as for any trust negotiation, $unlocked(C_{k+j+1}, C_{k+j})$ (recall that $C_k = \emptyset$))

Example: Figure 2.1 shows two separated strategies of ATN, which are Eager strategy and Parsimonious. They are descripted detail in Figure 2.3 and Figure 2.4, respectively. A ratio of disclosed policies over all policies call a policy disclosure rule, and a number of transmissions of message between two negotiators call a round of negotiation. For example, the eager strategy gives the consensus in the order as shown in Figure 2.3, compliant R, c_1, s_1. The disclosure rule is 6/8 = 0.75 and the negotiation finish with in 3 rounds. While parsimonious strategy is shown in Figure 2.4, the disclosure rule is 4/8 = 0.5 and the negotiation finish in 5 rounds.

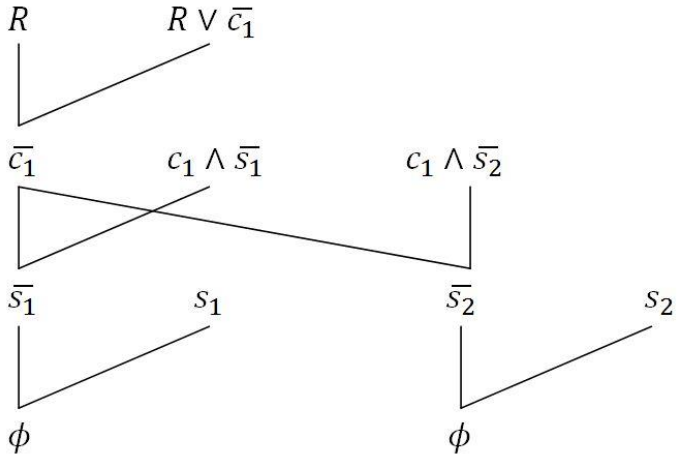

Figure 2.4: Example of Parsimonious strategy

2.2 E-learning

E-learning is learning utilizing electronic technologies to access educational curriculum outside of a traditional classroom. In most cases, it refers to a course, program or degree delivered completely online. Nowadays, e-learning make a vital impact to education system since the Internet and communication technologies have been developed rapidly. E-learning is developed in various point of aspect such as an educational purpose, medium for communication or education management information systems (EMIS). Mainly, it is designed to let people share information or electronic media over the Internet. Many universities offer students the online courses which lead to the need of mechanics to provide online services for example, authorization, networking and database.

Many organizations are making e-learning standards, such as, ADL, IMS, IEEE LTSC and so on. The IEEE Learning Technology Standards Committee (LTSC) [13] publishes many standards for e-learning, such as, Learning Technology Systems Architecture (LTSA), Learning Object Metadata (LOM), Content data model (CDM) and so on. Figure 2.5 show the LTSA system component which composed of

- Processes: learner entity, evaluation, coach, delivery
- Stores: learner records, learning resources
- Flows: learning parameters, behavior, assessment, information, learner information (three times), query, catalog info, locator (twice), learning content, multimedia, interaction context.

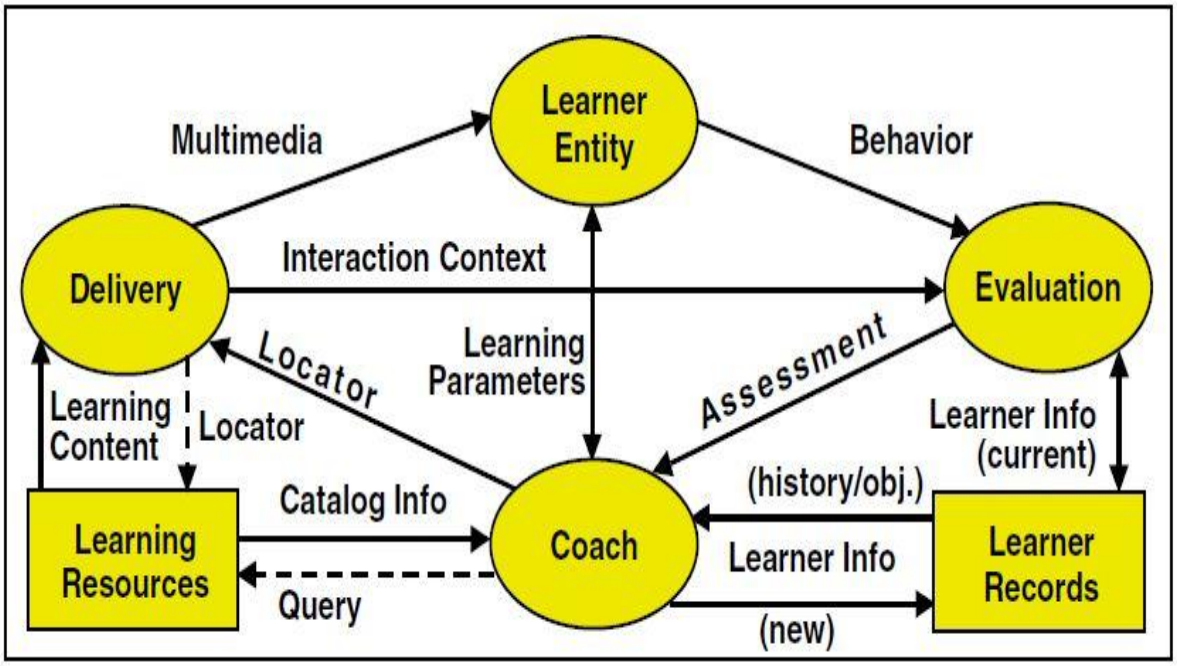

Figure 2.5: IEEE Standard for LTSA Architecture

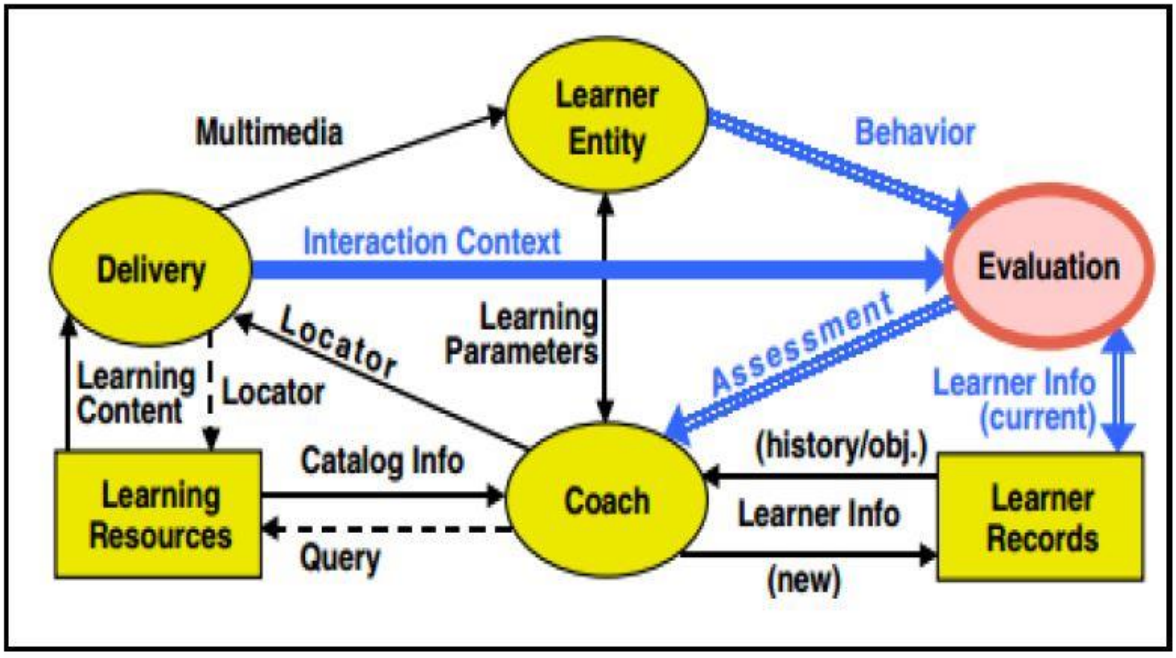

Figure 2.6: Evaluation Process of LTSA

Learner Entity represents the student which can be single student or a group of student. The Learner entity receives a multimedia data via the multimedia data flow and the behavior of the learner can be observed via the behavior data flow. Later on, the behavior information such as keyboard click, mouse click, written response, and so on will be used at evaluation process. The learning data flow is used for the learner to communicate with the coach.

Figure 2.6 shows the evaluation process where the behavior of the learner is used to produce measurement of the learner entity. The evaluation process requires the behavior of the learner information, the context to the learner's behavior. The output of the evaluation will be sent to the coach via the assessment data flow and the system keeps the learner's evaluation result via the learner information data flow. Which consist of Inputs/Outputs as following:

- Input: The learner entity's observable behavior via the behavior data flow.
- Input: The interaction context data flow may provide context to the learner entity's behavior to determine the appropriate evaluation.
- Output: Assessment information may be sent to the coach via the assessment data flow.
- Input/output: Learner information may be retrieved and stored (via the learner information data flow) during evaluation processing in the learner records.

Example: Let the learner chooses one best correction answer from the multiple choice question. The right answer is "#3". The evaluation process waits for the learner to input the answer which can be a key stroke of "3","#3", "three" which are the set of correct answer.

Generally, e-learning website composed of web-server, database, users and engine. IEEE standard for learning technology systems architecture composed of learning resource, learner records, and evaluation. Carchiolo et al. [14] proposed a general model for e-learning that consists of domain database, profiles of students and teachers, engine, course and feedback as shown in Figure 2.7.

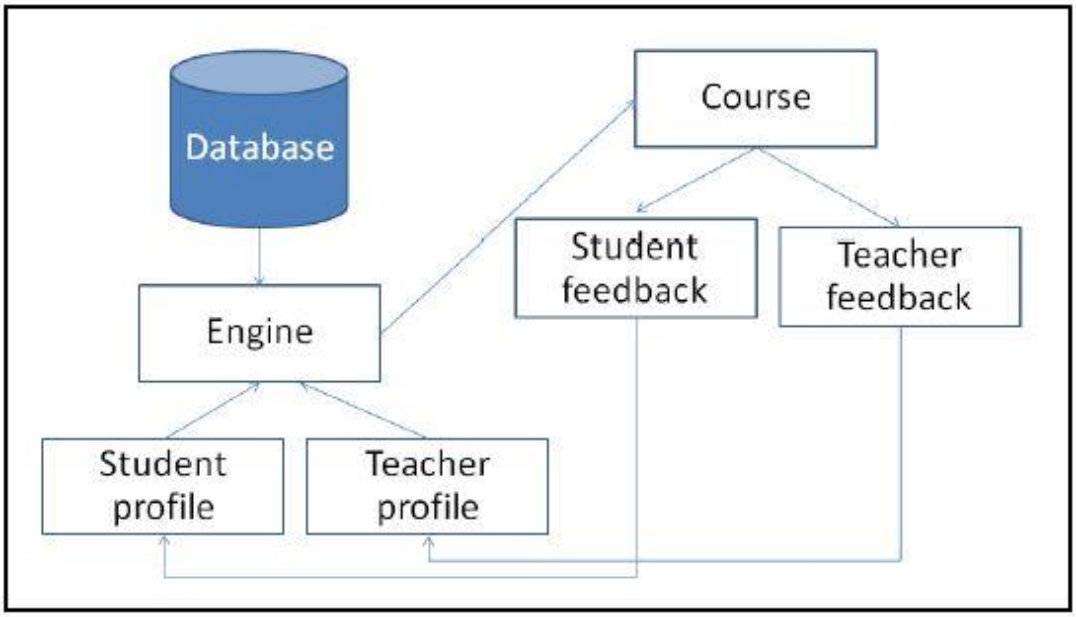

Figure 2.7: General model for e-learning

The domain database stores ordinary course information such as course number, name and period of learning time. Moreover, it contains links to the course material such as lesson, slides and so on. Normally, the structure for databases are graph-based [15-16]. The information of the teacher and student are stored without any cryptography strategy. Traditionally, core of e-learning system composed of a server such as Apache, a database and the engine. The lesson will be uploaded to the server and transfer to store in database by the teacher with PHP language. Students can login to learn or take a test online and send their answers to process result checking at the server then the score will be stored in the database.

Figure 2.8 show the e-learning system architecture, there are three types of client which are student, teacher and admin.

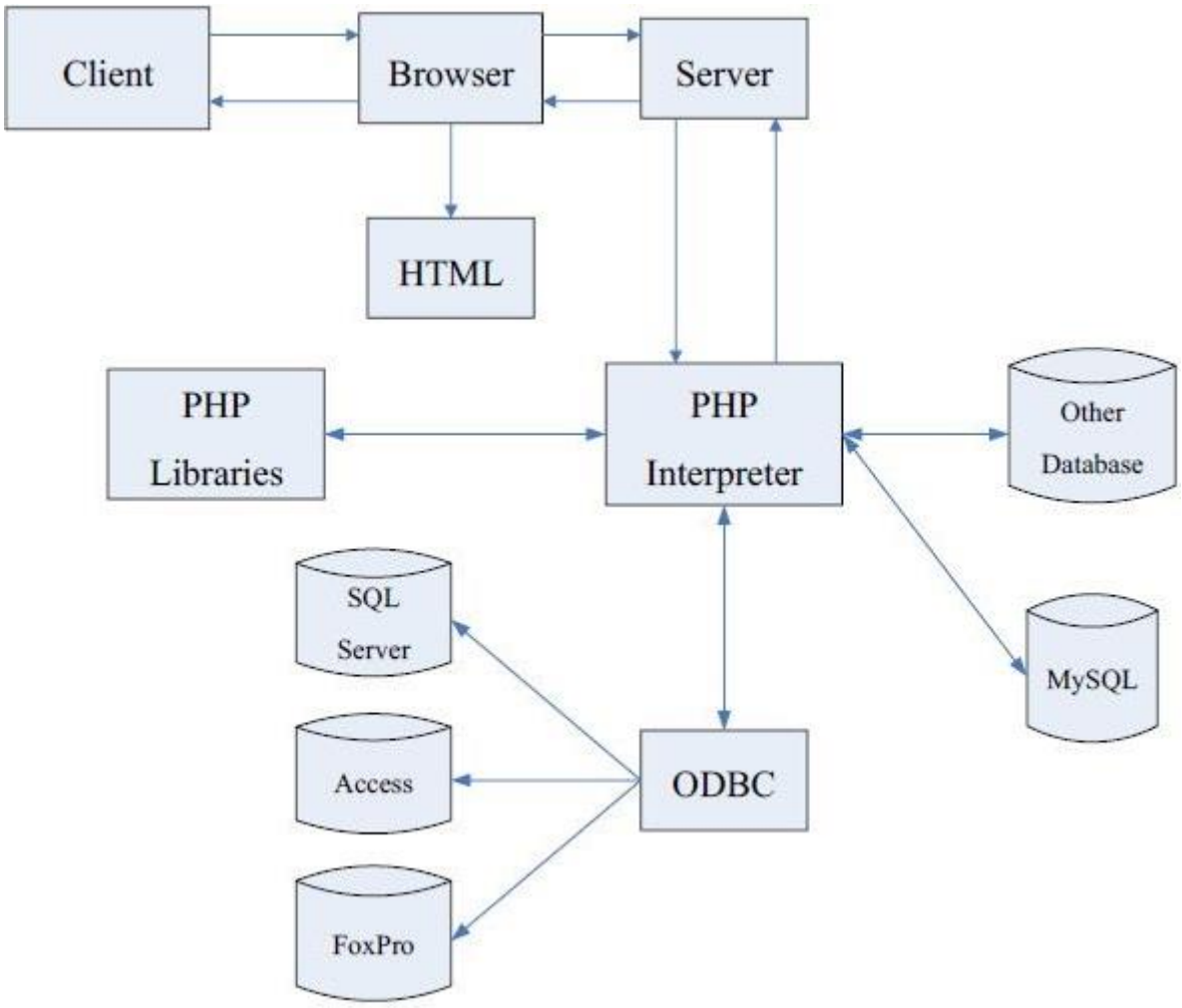

Figure 2.8: System architecture of e-learning

2.3 Cryptographic Tools

There are three kinds of cryptographic tools that have been used in this research:

- Paillier Crytosystem
- Homomorphic Public Key Encryption
- Private Matching

2.3.1 Paillier Cryptosystem

Pascal Paillier presents an asymmetric algorithm for public key cryptography which satisfies an additive homomorphic property, this scheme allow us to get key generation and decryption processes distributed among semi-trusted authorities sharing private key.

The Paillier cryptosystem [17] consists of three algorithms which are key generation, encryption, and decryption.

2.3.1.1 Key Generation

Let N be pq, a multiplication of large primes p and q, $g \in Z_{N^2}^*$ be generator whose order divides N. Compute as following:

$$\lambda = LCM(p-1, q-1) \tag{2.1}$$

$$\mu = (L(g^{\lambda}(mod\ n^2)))^{-1} \tag{2.2}$$

where L is defined by

$$L(u) = (u-1)/n. \tag{2.3}$$

The public key is (N, g) and the private key is (λ, μ).

2.3.1.2 Encryption

A ciphertext of M is defined with randomly chosen $r \epsilon Z_{N^2}^*$ as,

$$E(M) = g^M r^N (mod\ n^2) \tag{2.4}$$

2.3.1.3 Decryption

Given ciphertext c, plaintext M is compute by

$$L(c^{\lambda}(mod\ n^2)) \tag{2.5}$$

2.3.2 Homomorphic Public Key Encryption

A homomophic encryption scheme [17-18] is an encryption scheme in which the plaintext are taken from a group G. The homomorphic public key encryptions is building block using when the two group want to learn the sum of their corresponding plaintext without revealing the exactly value of each other's plaintext. Normally, this computation involves a modular multiplication of encryptions, as below:

$$E(a).E(b) = E(a + b) \tag{2.6}$$

It is easy to see that:

$$E(a)^c = E(a.c) \tag{2.7}$$

2.3.3 Private Matching

In EUROCRYPT 2004, Freedman, Nissim, and Pinkas [19] presented a basic private matching protocol (FNP04) in the model of semi-honest adversary. The FNP04 protocol is asymmetric, which means these two parties will not know the same information at any point in the protocol. In the protocol, both parties are assumed to act according to their prescribed actions as shown in Figure 2.9

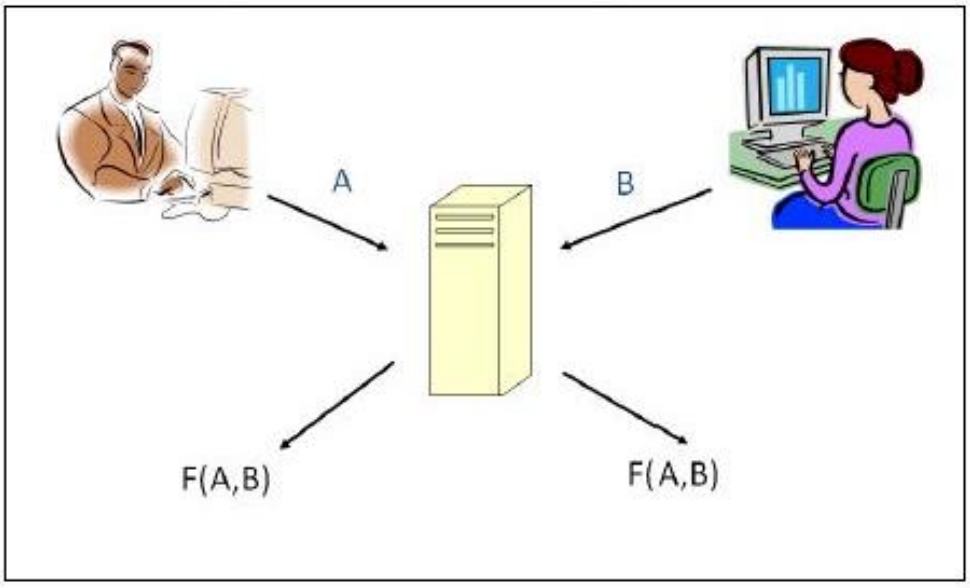

Figure 2.9: Private matching

A private matching scheme is a two group protocol between a client (chooser) C and server (sender) S. Input of client is a set of inputs of size k_c, drawn from some domain of size N; Server's input is a set of size k_s drawn from same domain. At the conclusion of the protocol, C learns which specific inputs are shared by both C and S. That is, if C inputs $X = \{x_1, x_2, \dots, x_{k_c}\}$ and S inputs $Y = \{y_1, y_2, \dots, y_{k_s}\}$. The client uses a polynomial having element of X as its root defined as

$$\begin{aligned} P(x) &= (x - x_1)(x - x_2) \dots \left(x - x_{k_c}\right) \\ &= a_0 + a_1 x + \cdots + a_k x^k \end{aligned} \tag{2.8}$$

to encode X and then send to S a sequence of ciphertexts

$$\begin{aligned} E\big(P(x)\big) &= E(a_0 + a_1 \cdot x^1 + \cdots + a_k \cdot y^k) \\ &= E(a_0) \cdot E(a_1)^{x1} \dots \cdot E(a_k)^{yk} \end{aligned} \tag{2.9}$$

For, server S compute

$$\left(\prod_{i=0}^{k_c}(E(a_i))^{y^i}\right)E(y) = E(P(y)^r \cdot E(y)$$

$$= E(r \cdot P(y) + y) \qquad (2.10)$$

and sends k_s ciphertexts to C in random order, where r is uniform random number.

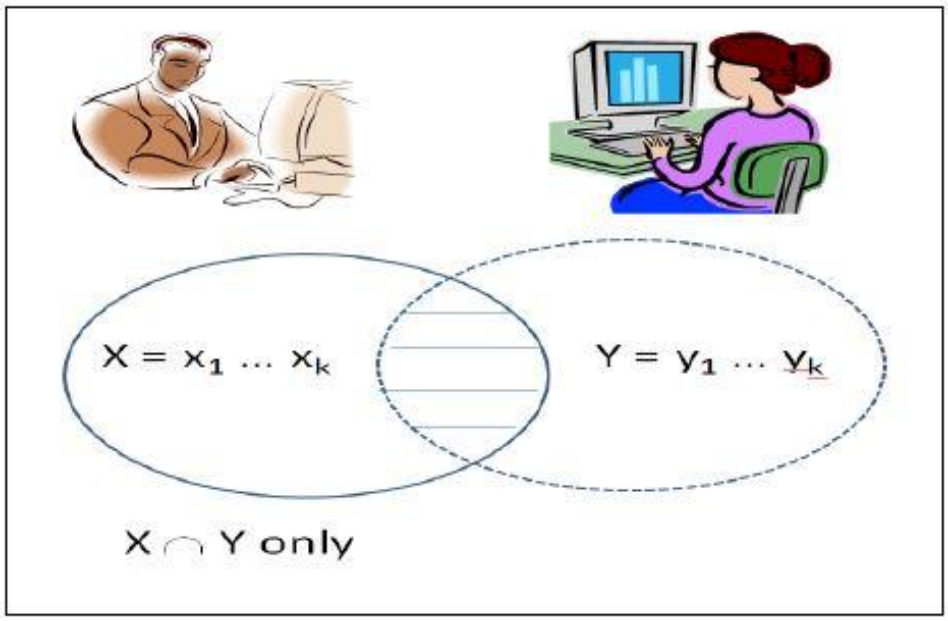

Figure 2.10: Private matching example

Finally, client C decrypts the ciphertexts to obtain the elements of the intersection $X \cap Y$ without learning any other element as shown in Figure 2.10

CHAPTER 3

METHODS AND TOOLS

Our proposed method is presented in this chapter. The proposed method is divided into two parts. First part, eager strategy and parsimonious strategy are simulated. The second part, the score computation system is designed.

3.1 Contribution of Strategies Simulation

Two strategies are simulated with conjunctive and disjunctive models: eager strategy and parsimonious strategy

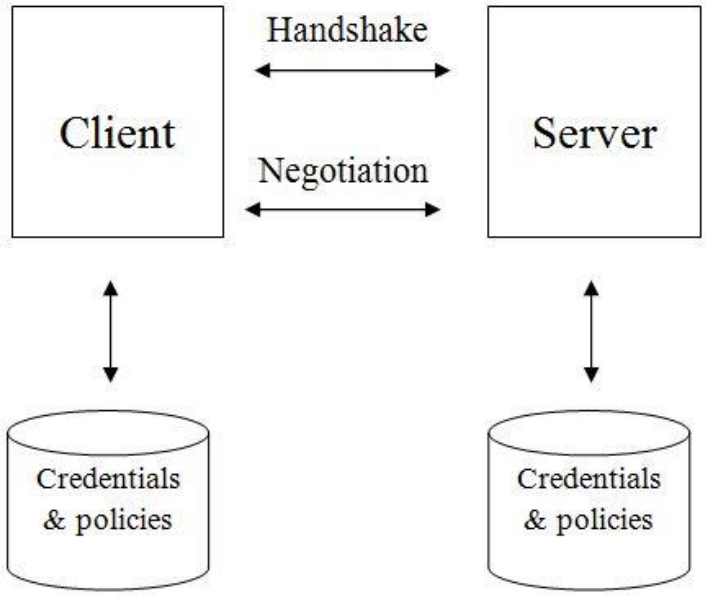

Figure 3.1: The system architecture of trust negotiation

Figure 3.1 is the system architecture of trust negotiation that is composed of the client, the server, credentials and policies of the client and the server. Firstly, we have to select the height of the tree and the strategy at handshake process, then negotiation starts. The client

handles access control policies and the credentials of the client that governs disclosure of those credentials by verifying, credentials from the client are submitted the service governing policy before forwarding the request to the server. The server exchanges by disclosing all available server credentials to the client, until a client requests a service without submitting enough credentials to gain authorized access.

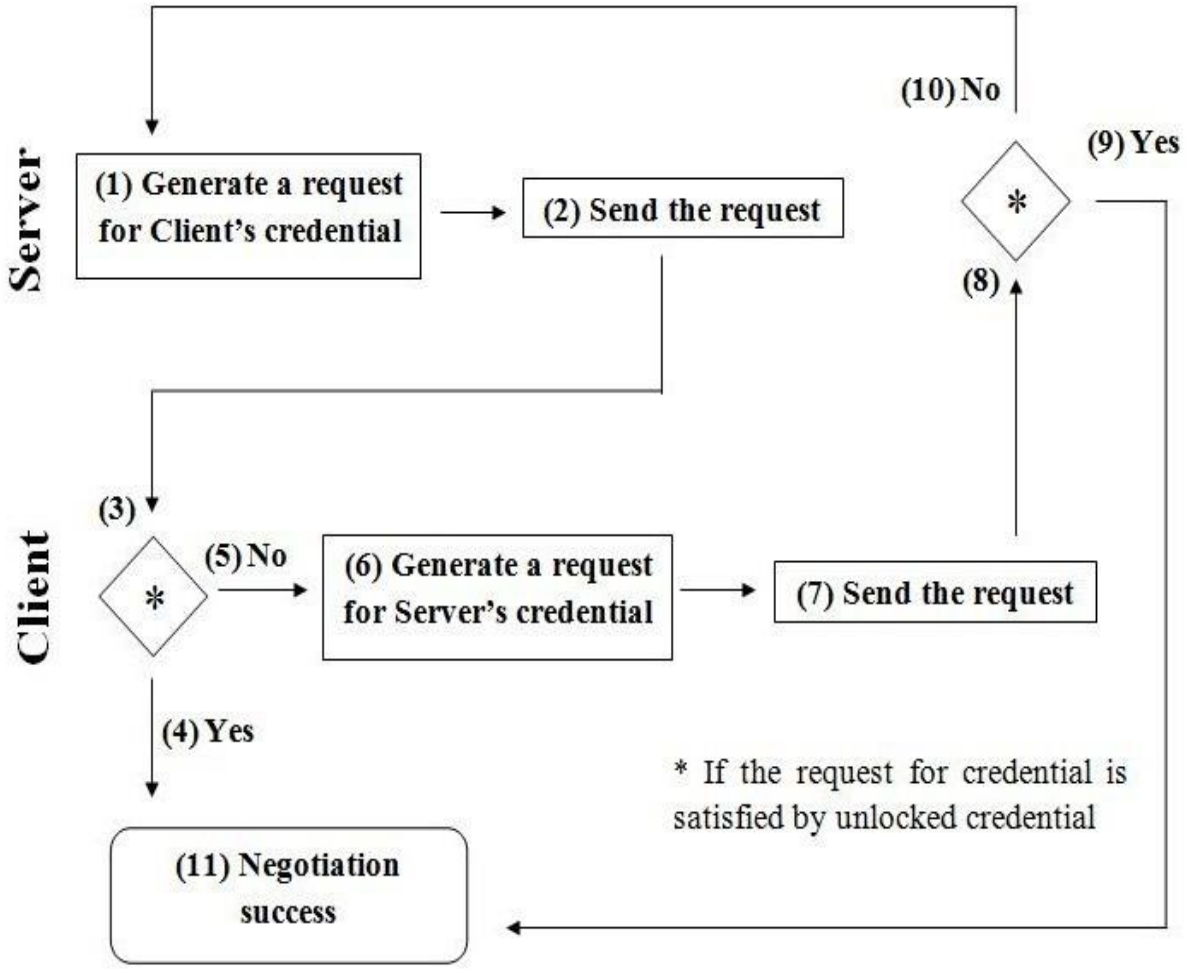

Figure 3.2: Exchange of credentials

The goal is to experiment eager strategy and parsimonious strategy with conjunctive and disjunctive model. The process of credential exchange is shown in Figure 3.2.

(1) The server generate a request for client's credentials

(2) The server sends server's request to the client

(3) The client specifies whether the server's request is satisfied by the unlocked credentials from the client.

(4) If the server's request is satisfied by unlocked credentials from the client, go to (11).

(5) If the server's request is not satisfied by unlocked credentials from the client, go to (6).

(6) The client generates a request for server's credentials.

(7) The client sends client's request to server.

(8) The server specifies whether the client's request is satisfied unlocked credentials from the server.

(9) If the client's request is satisfied by unlocked credentials from the server, go to (11).

(10) If the client's request is not satisfied by unlocked credentials from the server, go to (1)

(11) The server and the client find the sequence to exchange credentials by inversion previous request for credentials and exchange credentials in accordance with it, the negotiation finishes in success.

3.2 Score Computation

The overall framework is shown in Figure 3.3. There are three types of users which are admin, teacher and student. The teacher can set up both subjective and objective questions. The proposed scheme is applied on the subjective question. The teacher can choose whether he would like to use the scheme or not. In Mathematical question, generally, there is exactly one solution to the question so the proposed scheme is not required.

This scheme is applied on a subjective question with many answers in one question. For example, what are the seasons in the America? The answers are spring, summer, fall, and winter where the order or position of the answer can be shuffled. The full marks of this question are ten. Suppose that there are four students Alice, Bob, Carol and David. Alice answers spring, fall, winter and snow. Bob answers summer, winter, hot and sun. Carol answers spring, spring, spring and spring. David answers winter, fall, summer and spring.

The score is compute as these steps:

1) Fetches the answer from the teacher and students. Extract the answer sentence into word.

2) Transform each word to vector and create a dictionary table.

3) Compute the similarity and score with inner product.

A dictionary table index will be created based on a teacher's answer; in this case, there are 4 indexes as shown in Table 3.1. In transformation process from word to a vector space, the frequency value must be normalize depends on the answer key; in this case, it must be reduced to 1. After the transformation, the inner product calculated as;

- Alice = teacher's answer x Alice's answer = (1*1) + (1*0) + (1*1) + (1*1) = 3
- Bob = teacher's answer x Bob's answer = (1*0) + (1*1) + (1*0) + (1*1) = 2
- Carol = teacher's answer x Carol's answer = (1*1) + (1*0) + (1*0) + (1*0) = 1
- David = teacher's answer x David's answer = (1*1) + (1*1) + (1*1) + (1*1) = 4

The score for each student's score can be computed as

- Alice = (3/4)*10 = 7.5
- Bob = (2/4)*10 = 5
- Carol = (1/4)*10 = 2.5
- David = (4/4)*10 = 10

Table 3.1: A dictionary table for the example

index	word
1	**spring**
2	**summer**
3	**fall**
4	**winter**

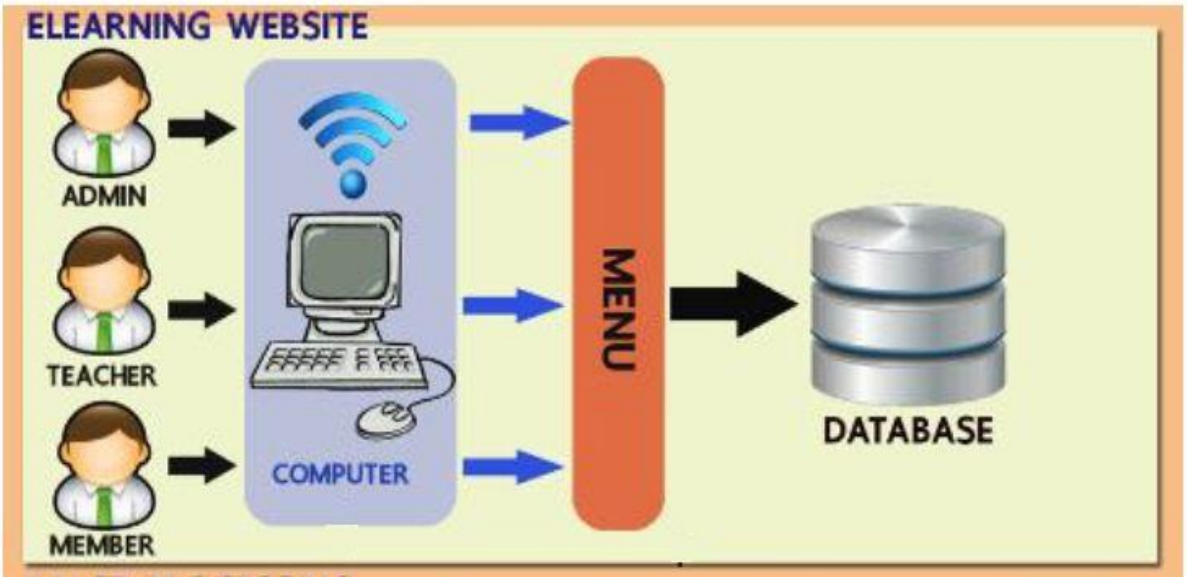

Figure 3.3: Overall score computation of system framework

CHAPTER 4
EXPERIMENT RESULTS

This chapter addresses the results of our research on privacy-preserving base automated trust negotiation in e-learning system.

4.1 Experiment of Strategies

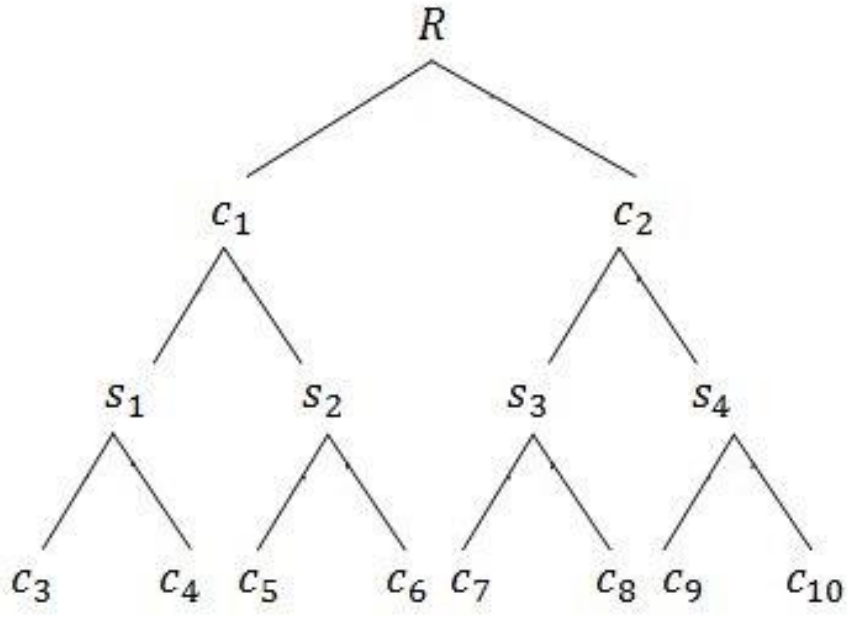

Figure 4.1: Example of all disjunctive policies using the complete binary tree

Firstly, the height of binary tree is arranged from one to thirteen and the program automatically generated credentials that matched with the height of the tree. Secondly, the time is recorded while negotiators begin exchanging their credentials. The parent nodes credentials are locked by their child nodes and the leaves credentials are unlocked. Lastly, both strategies have simulated in conjunctive and disjunctive model. Figure 4.1 shows the example of disjunctive policies by using the complete binary tree that height of tree is three, while Figure

4.2 is the example of conjunctive policies. First row contains the client's credentials c_1 and c_2, second row contains the server's credentials $s_1, \dots, s_4$, and third row contains the client's credentials $c_3, \dots, c_4$. The second row credentials are locked by the third row credentials, and credentials of first row are locked by the second row credentials while credentials of the third row are unlocked.

Example: c_1 is locked by $s_1 \vee s_2$, represented by $c_1 \rightarrow s_1 \vee s_2$ as in Figure 4.1, and in Figure 4.2 can be write as c_1 is locked by $s_1 \wedge s_2$, represented by $c_1 \rightarrow s_1 \wedge s_2$

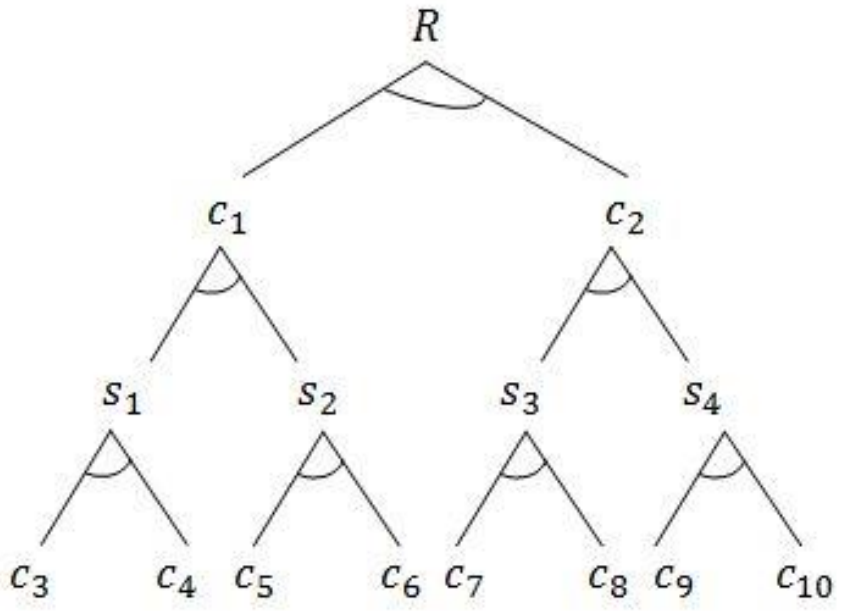

Figure 4.2: Example of all conjunctive policies using the complete binary tree

From Figure 4.2. The process of our example is shown in Table 4.1. The target of negotiation is R. Firstly, the client requested R and the server requested $c_1 \wedge c_2$ from the client. The client checked that $c_1 \wedge c_2$ was locked by the server's credentials $s_1, \dots, s_4$so the client requested $s_1, \dots, s_4$ from the server. In order to be able to give $s_1, \dots, s_4$ to the client, the server needed $c_3, \dots, c_{10}$ so the server requested $c_3, \dots, c_{10}$ from the client. The client checked access control policies for $c_3, \dots, c_{10}$ and found that these credentials was unlocked so the client sent $c_3, \dots, c_{10}$ to the server. After that the server used received credentials to unlock $s_1, \dots, s_4$ then sent them back to the client. Then the client can unlock $c_1 \wedge c_2$ and sent them to the server.

Finally, the server received the client credentials and got R, and sent it to the client, the negotiation is success. Where round of negotiation is eight and disclosure rules are seven.

Table 4.2 shows the example of parsimonious strategy in all disjunctive policies using the complete binary tree that height of the binary tree is three. The target of negotiation is R, Firstly, the client requested R, in order to get R the server need $c_1 \vee c_2$, the server requested only c_1 from the client. The client checked that c_1 was locked by the server's credentials $s_1 \vee s_2$ so the client requested s_1 from the server. In order to be able to give s_1 to the client, the server needed $c_3 \vee c_4$ so the server requested c_3 from the client. The client checked access control policies for c_3 and found that these credentials was unlocked so the client sent c_3 to the server. After that the server used c_3 to unlock s_1 then sent it back to the client. Then the client can unlock c_1 and sent it to the server. Finally, the server received the client credentials and got R, and sent it to the client, the negotiation is success .Where disclosure rules are three and number of round is eight.

Table 4.1: Example of negotiation process of parsimonious strategy for all conjunctive policies

Client	Server
Request R →	
	← Request $c_1 \wedge c_2$
Request $s_1 \wedge s_2$ and $s_3 \wedge s_4$ →	
	← Request $c_3 \wedge c_4$, $c_5 \wedge c_6$, $c_7 \wedge c_8$ and $c_9 \wedge c_{10}$
Sent $c_3 \wedge c_4$, $c_5 \wedge c_6$, $c_7 \wedge c_8$ and $c_9 \wedge c_{10}$ →	
	← Sent $s_1 \wedge s_2$ and $s_3 \wedge s_4$
Sent $c_1 \wedge c_2$ →	
Success get R	

Table 4.2: Example of negotiation process of parsimonious strategy for all disjunctive policies

Client	Server
Request R →	
	← Request c_1
Request s_1 →	
	← Request c_3
Sent c_3	
	← Sent s_1
Sent c_1 →	
Success get R	

Figure 4.3 shows the processing time of eager strategy and parsimonious strategy. The processing time of both strategies grow up following the increasing of tree's height. In Figure 4.4, the number of round is compared between eager strategy and parsimonious strategy. The result shows that the number of round is increase in both strategies.

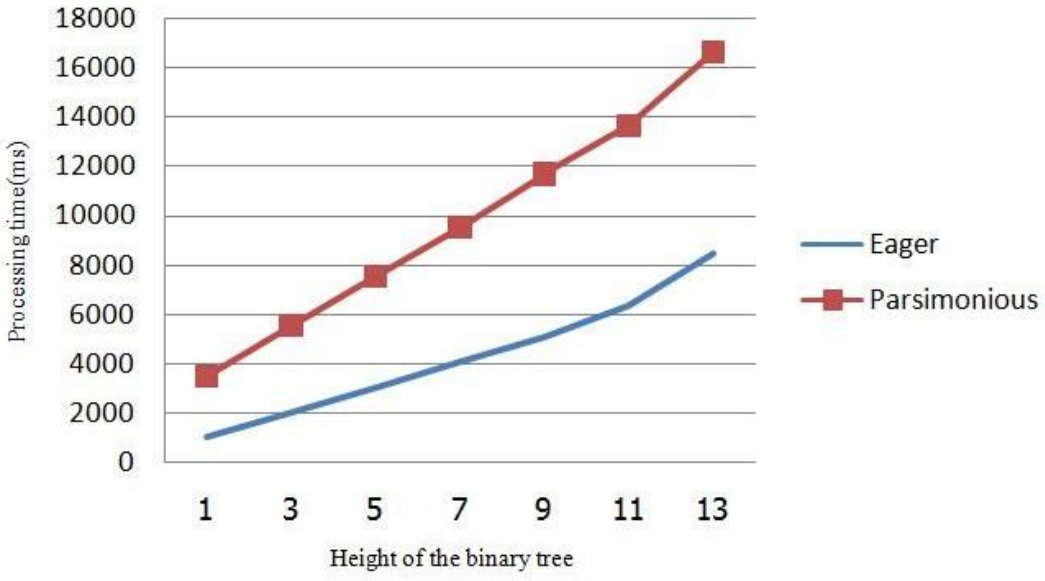

Figure 4.3: Processing time of eager and parsimonious strategies

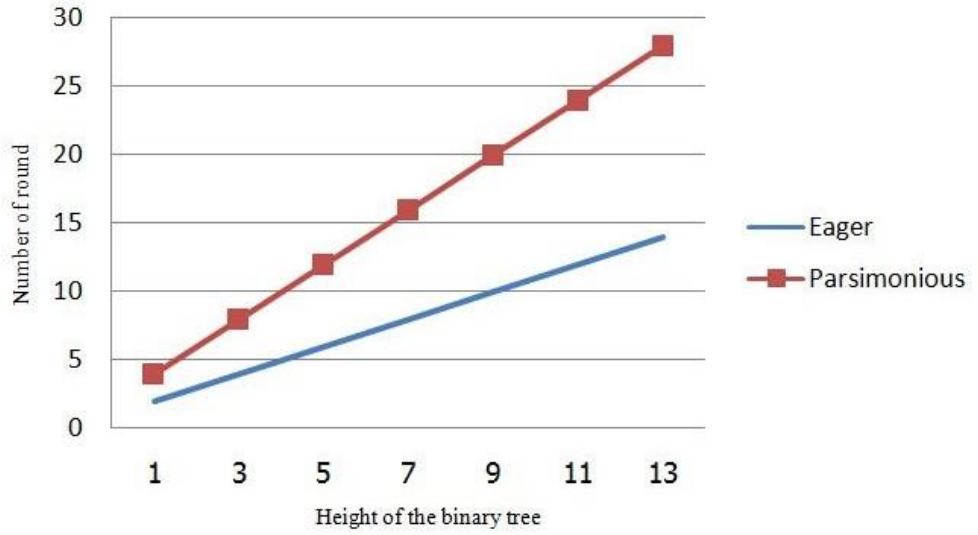

Figure 4.4: Number of round

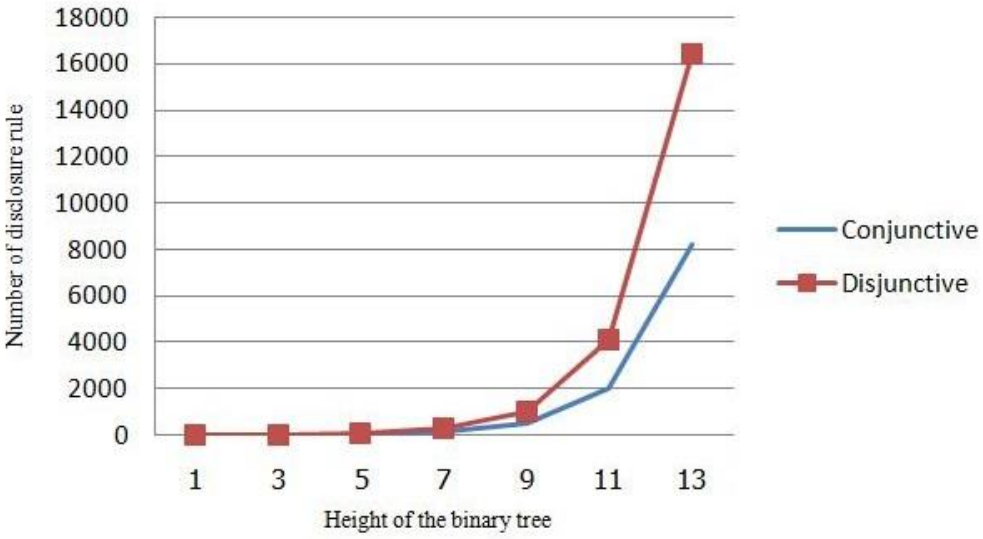

Figure 4.5: Number of disclosure rule of eager strategy

Figure 4.6: Number of disclosure rule of parsimonious strategy

Figure 4.5 and Figure 4.6 show the comparison between conjunctive and disjunctive in the number of disclosure rule, respectively eager strategy and parsimonious strategy. In eager strategy, the number of disclosure rule of disjunctive is higher than conjunctive rule. In parsimonious strategy, the number of disclosure rule of conjunctive case is higher than the disjunctive case.

4.2 Application of Privacy-Preserving based on Automated Trust Negotiation

4.2.1 Authentication

In authentication process, the student and the teacher exchange their credentials, the target of the student is to takes a test. Before takes a test, the student must satisfy by the teacher that he already register to the course by sent the registration information to the teacher. The registration information contains content of the student's credentials such as name, address, course ID, and so on. The registration information is provided by the general register office (Third Trust Party). Table 4.3 and Table 4.4 show the student's credentials and teacher's credentials respectively.

Table 4.3: The credentials of the student

Student	
Student ID	Name
Personal ID Card	Password
Address	Birthday
Telephone	Register Course

Table 4.4: The credentials of the teacher

Teacher	
Teacher ID	Name
Personal ID Card	Password
Address	Birthday
Telephone	Position

For example: the student want to take a test, before takes a test, the student need to authentication by the teacher, so the target is authentication to the course represented by R. in order to get the R, the student need to show the student ID and course ID represented by $c_1 \wedge c_2$, while course ID is belong to the teacher ID represented by s_1 (In order to know the course that he want to takes, he must know the teacher who teach). The trust target graph show as below in Figure 4.7. Table 4.5 is show the negotiation process of authentication. First the student request R, in order to get R, the teacher need the student's credentials $c_1 \wedge c_2$ the sent request to the student. At this point c_1 is unlock while c_2 is lock by s_1 so the student request s_1 from the teacher. The teacher sent teacher's credentials s_1 to the student, after that student use received credential to unlocked c_2 and sent $c_1 \wedge c_2$ to the teacher. Then negotiation success.

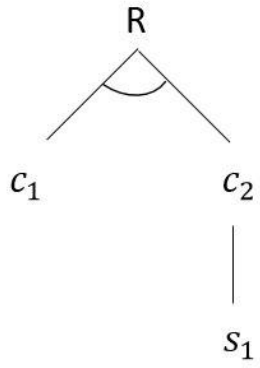

Figure 4.7: Trust target graph of authentication

Table 4.5: The negotiation process of authentication

Student	Teacher
Request R →	
	← Request $c_1 \wedge c_2$
Request s_1 →	
	← Sent s_1
Sent $c_1 \wedge c_2$ →	
Success get R	

4.2.2 Privacy-Preserving E-learning in Score Computation

First, student and teacher must register for the account. Then, teacher can create lesson and question using building tools implement with PHP. Student can login and takes a pretest. After finish taking the test, the student's answer will be encrypted using Paillier cryptosystem and send to the teacher. The teacher compute her answer key with the encrypted message received from the student, shuffle and send back the computed message to the student. The student decrypts the messages and gets "1" for the correct answer and gets random number for the wrong answer. The system can be placed in LTSA system as shown in Figure 4.8. In the proposed scheme, the learner and coach communicate directly to each other. When the student

wants to take a pre-test, the student can directly request questions from the teacher. After finish answering, the student encrypts his set of answer with his public key using Paillier cryptosystem and sends ciphertexts to the teacher. The teacher puts the correct answer in each ciphertexts, shuffle and send back the ciphertexts to the student. Finally, the student decrypts the ciphertexts and gets "1" for the correct answer or gets "random" for the wrong answer.

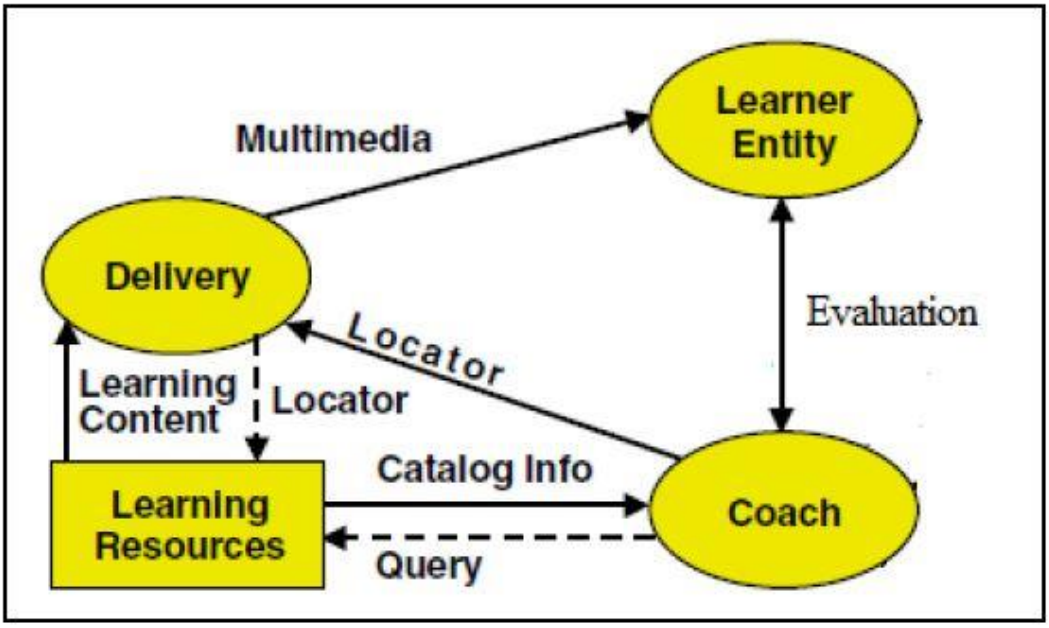

Figure 4.8: The system place in LTSA

Our scheme performs in three phases: encrypting phase, blind matching phase and unbinding phase. The encrypting phase will be performed by the student in order to encrypt his answer and sent the ciphertext to the teacher. The blind matching phase will be performed by the teacher to securely put her answer key into the ciphertext she received and send them back to the student. The unbinding phase will be performed by the student to check for the result of his test by decrypting the received ciphertext.

The student has his set of answers. $S = \{s_1, s_2, \dots, s_n\}$ is a set of n answers.

For example, the student answers 3 questions then $S = \{s_1, s_2, s_3\}$. The teacher has a set of her answer keys T where $T = \{t_1, t_2, \dots, t_n\}$. The answer s_n corresponding to the key t_n. Let E_s and D_s be public key encryption and decryption for student respectively.

Our proposed scheme performs as follow;

1) The student generates his own public key, private key and parameters for a semantically-secure homomorphic encryption and decryption scheme.
2) Encrypting phase:

   The student encrypts his answer using his public key and sends all encrypted messages, $E_s(P(x))$ to the teacher.
3) Blind matching phase:

   The teacher picks fresh random number r then evaluates the encrypted polynomial, $E_s(rP(b)+1)$, shuffle and send back to the student.
4) Unbinding phase:

   The student decrypts the received message with his private key wishing have $D_s\big(E_s(rP(b)+1)\big)=1$ which means his answer is correct.

Table 4.6: Sample negotiation process

Student	Teacher
Question 1 : a Question 2 : b Question 3 : d	Question 1 : a Question 2 : c Question 3 : d
$A1=E_s\big(P_1(x)\big)$ $A2=E_s\big(P_2(x)\big)$ $A3=E_s\big(P_3(x)\big)$	
	$B1=E_s(rP_1(t_1)+1)$ $B2=E_s(rP_3(t_3)+1)$ $B3=E_s(rP_3(t_3)+1)$ ← Shuffle and send $B3, B1, B2$
Decrypt $B3, B1, B2$ then get $B1=1$ $B2=$ random $B3=1$	

Example: There are 3 questions in a test. The student's answer is a, b and d while the teacher's key is a, c and d respectively. The student begins with encrypting his answers one by one so he gets A1, A2 and A3 then send them to the teacher respectively. The teacher puts the key in the ciphertext using Homomorphic properties and private matching. The teacher does not want the student to learn which question he did correct or incorrect so she shuffles sequence of the messages then sends them back to the student. The student decrypts and gets either 1 for the correct answer or gets random number for the wrong answer. At this point, the student learns only how many points he gets but cannot learn exactly which question he did right or wrong as shown in Table 4.6.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Comparison the Strategies

In this section, the simulation and comparison of the eager strategy and parsimonious strategy in conjunctive case and disjunctive case are presented. The results demonstrate the strength and weakness of both strategies. Eager strategy is disclosed all unlocked and unrelated credentials. The processing time of eager strategy is faster than parsimonious strategy. Parsimonious strategy is disclosed only relevant credentials and many rounds of communication with less disclosure credentials. Hence, eager strategy is appropriate for exchanging non sensitive credentials that focus on less communication cost, while parsimonious strategy work well with sensitive credentials.

5.2 Privacy in E-learning System

In this thesis examines secret exchange on the score computation in e-learning system to protect both the student's privacy and the teacher's sensitive information. A crypto-graphical protocol has proposed for computing score of student with full privacy preserved, which performs in three phase which are encrypting phase, blinding phase and unbinding phase. Our protocol allows only the student learns his score of the test without disclosing the answer key while the teacher and server learn nothing.

5.3 Future Work

We plan to simulate hybrid strategy, design automated trust negotiation method to avoid the cyclic dependency and experiment proposed method with automated trust negotiation.

REFERENCES

[1] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp. 164–173. IEEE Computer Society Press, May 1996.

[2] L. Ronald. Rivest and B. Lampson, "SDSI - A simple distributed security infrastructure," *http://theory.lcs.mit.edu/~rivest/sdsi11.html*, October 1996.

[3] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI certificate theory," *IETF RFC 2693,* September 1999.

[4] D. Clarke, J. E. Elien, C. Ellison, M. Fredette, A. Morcos, and L Ronald, R. Rivest, "Certificate chain discovery in SPKI/SDSI," *Journal of Computer Security*, 9(4): pp. 285–322, 2001.

[5] N. Li, W. H. Winsborough, and J. C. Mitchell, "Distributed credential chain discovery in trust management," *Journal of Computer Security*, 11(1): pp. 35–86, February 2003.

[6] W. H. Winsborough, K. E. Seamons, and V. E. Jones, "Automated trust negotiation," *DARPA Information Survivability Conference and Exposition*, volume I, pp. 88–102, IEEE Press, January 2000.

[7] K. E. Seamons, M. Winslett, and T. Yu, "Limiting the disclosure of access control policies during automated trust negotiation," *Proceedings of the Symposium on Network and Distributed System Security*, February 2001.

[8] W. H. Winsborough and N. Li, "Towards practical automated trust negotiation," *Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks*, pp. 92–103, IEEE Computer Society Press, June 2002.

[9] T. Yu, M. Winslett, and K. E. Seamons, "Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation," *ACM Transactions on Information and System Security*, 6(1): pp. 1–42, February 2003.

[10] T. Yu and M. Winslett, "Unified scheme for resource protection in automated trust negotiation," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 110–122. IEEE Computer Society Press, May 2003.

[11] Yuan, E and J. Tong, "Attributed based access control (ABAC) for Web services" *Proceeding of IEEE International Conference on Web Services (ICWS'05)*, 2005

[12] K. E. Seamons, M. Winslet, and T. Yu, "Protecting privacy during online trust negotiation*," Proc. of the 2nd Workshop on Privacy Enhancing Technologies, Springer-Verlag*, pp. 129-143, 2003,

[13] IEEE standard for learning technology-learning technology systems architecture (LTSA) IEEE Std 1484.1-2003, *IEEE Std 1484*, pp. 01-97, 2003.

[14] V. Carchiolo, A. Longheu, M. Malgeri, "Dynamic web pages for tuning formative path", *DIIT Technical Report #1013, University of Catania*.

[15] C. Chiu, "The Authority Structure Problem of Computer Supported Collaborative Concept Mapping System for Elementary Student", *IEEE International Conference on Advanced Learning Technologies (ICALT2001), Madison, WI, USA, Los Alamitos, CA: IEEE Computer Society*, pp. 57-56, August 6-8, 2001

[16] Y. Fujiwara, B. Matsuzawa and S. Okada, "Learning Assistance Expert System with a Self-Adaptive Function Based on a Causal Network," *The World Multiconference On Systemics, Cybernetics And Informatics,* July 22 - 25, 2001, Orlando, Florida, USA.

[17] P. Paillier: "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," *EUROCRYPT'99, Springer LNCS*, pp. 223-238, May 1999.

[18] T. Okamoto, S. Uchiyama, and E. Fujisaki, "Epoc: Efficient probabilisticpublic-key encryption," *IEEE P1363: Protocols from other families of public-key algorithms*, November 1998.

[19] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," *Eurocrypt 2004, Springer LNCS*, pp. 1-19, May 2004

APPENDIX

LIST OF PUBLICATIONS

Some parts of this work are published in the following articles.

International Conference Proceedings

1. Malanh Phetsavong, Pikulkaew Tangtisanon “Privacy-Preserving E-learning in Score Computation” International Technical Conference on Circuit/Systems Computer and Communications (ITC-CSCC 2014), Phuket, Thailand, 1 - 4 July 2014.
2. Malanh Phetsavong, Pikulkaew Tangtisanon “The Research on Eager Strategy and Parsimonious Strategy Simulate” Regional Conference on Computer and Information Engineering (RC-CIE 2014), Yogyakarta, Indonesia, 7 – 8 October 2014

Privacy-Preserving E-learning in Score Computation

Mr. Malanh Phetsavong
Department of International College
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand

Dr. Pikulkaew Tangtisanon
Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
ktpikulk@kmitl.ac.th

***Abstract*— The Internet technology has been developing very rapidly so anyone can get access to the Internet from anywhere in anytime. Many universities offer online courses where students learn take examination and get a degree online. There are tremendous of research focusing on e-learning frameworks to improve learning system with enhancement of new technologies while the needs of the user privacy have been ignored. This paper examines secret exchanging on the examination system to protect both a students' privacy and a teacher's sensitive information. Our protocol can be run without the Third Party. At the end of the protocol, the teacher learns nothing from the students and the students learn only scores they get not the answer keys.**

Keywords—E-learning; Information security; Cryptography; Privacy-preserving component;

I. INTRODUCTION

Nowadays, e-learning make a vital impact to education system since the Internet and communication technologies have been developed rapidly. E-learning is developed in various point of aspect such as an educational purpose, medium for communication or education management information systems (EMIS). Mainly, it is designed to let people share information or electronic media over the Internet. Many universities offer students the online courses which lead to the need of mechanics to provide online services for example, authorization, networking and database. Most e-learning researched have focused on course construction design and management while privacy of users have been ignored.

Many organizations are making e-learning standards, such as, ADL, IMS, IEEE LTSC and so on. The IEEE Learning Technology Standards (LTSC) [1] publishes many standards for e-learning, such as, Learning Technology Systems Architecture (LTSA), Learning Object Metadata (LOM), Content data model (CDM) and so on. Fig. 1 show the LTSA system component which composed of (1) Processes: learner entity, evaluation, coach, delivery, (2) Stores: learner records, learning resources, and (3) Flows: learning parameters, behavior, assessment information, learner information, query, catalog info, locator, learning content, multimedia, and interaction context. Learner Entity represents the student which can be single student or a group of student. The Learner entity receives a multimedia data via the multimedia data flow and the behavior of the learner can be observed via the behavior data flow. Later on, the behavior information such as keyboard click, mouse click, written response, and so on will be used at evaluation process. The learning data flow is used for the learner to communicate with the coach. Fig. 2 shows the evaluation process where the behavior of the learner is used to produce measurement of the learner entity. The evaluation process requires the behavior of the learner information, the context to the learner's behavior. The output of the evaluation will be sent to the coach via the assessment data flow and the system keeps the learner's evaluation result via the learner information data flow. For example, let the learner chooses one best correction answer from the multiple choice question. The right answer is "#3". The evaluation process waits for the learner to input the answer which can be a key stroke of "1","#1", "one" which are the set of correct answer.

Carchiolo et al. [2] proposed a general model for e-learning that consists of domain database, profiles of students and teachers, engine, course and feedback as shown in Fig. 3.

The domain database stores ordinary course information such as course number, name and period of learning time. Moreover, it contains links to the course material such as lesson, slides and so on. Normally, the structure for databases is graph-based [3, 4]. The information of the teacher and student are stored without any cryptography strategy. Traditionally, core of e-learning system composed of a server such as Apache, a database and the engine. The lesson will be

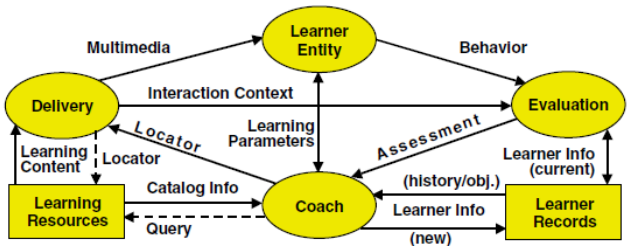

Fig. 1. IEEE Standard for LTSA Architecture.

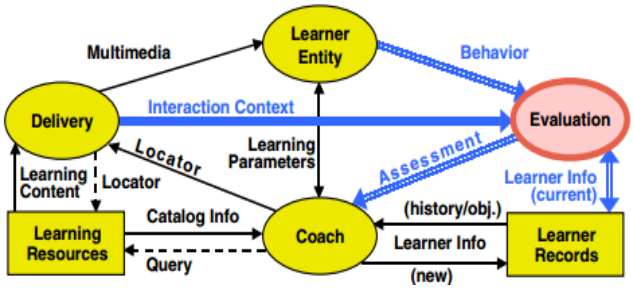

Fig. 2. Evaluation Process of LTSA.

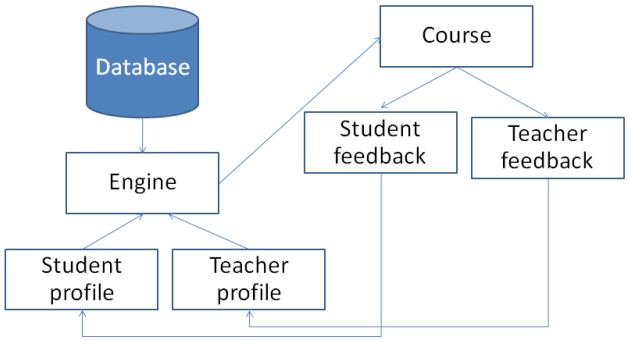

Fig. 3. General model for E-learning.

uploaded to the server and transfer to store in database by the teacher with PHP language. Students can login to learn or take a test online and send their answers to process result checking at the server then the score will be stored in the database. In previous works [5,6,7], the student's profile include information of the student such as name, surname, the student's available time, the preference of types of course and so on. This information is not stored in ciphertext so there are chances that the information will be hacked, as a consequence, the student's privacy is infringed.

In the case that the pre-test and post-test contain the same set of question to measure the improvement of the student, the celebrity student may not want to reveal his result of the pre-test to anyone including the teacher while the teacher is not willing to share answer key to the student since she must use this test again at the end of the course for the post-test.

Contributions: In this paper, we present a new scheme to protect a privacy of a student without disclosing keys from a teacher. Our proposed scheme satisfied the following properties: (1) The student will be able to get his test score without learning the answer key, (2) The teacher does not have to disclose her answer key to anyone even to the server, and Neither the teacher nor the server learns the student test result.

II. PAILLIER CRYPTOSYSTEM

Pascal Paillier presents an asymmetric algorithm for public key cryptography which contains an additive homomorphic cryptosystem property. This scheme works under three phases: (1) Key generation, (2) Encryption, and (3) Decryption. Paillier cryptosystem provides semantic security against chosen-plaintext attacks (IND-CPA). Many applications have been run with Paillier, for example, Electronic voting, Electronic cash and so on.

III. HOMOMORPHIC PUBLIC-KEY ENCRYPTION

To preserve the privacy of the students, we applied the public key cryptosystem (E) that satisfied the additive Homomorphic Public-key encryption such as Modified Elgamal and Paillier cryptosystem [8]. The homomorphic public key encryption is a building block using when the two players want to learn the sum of their corresponding plaintext without revealing the exactly value of each other's plaintext.

$$E[M_1]E[M_2] = E[M_1 + M_2] \quad (1)$$

$$E[M_1]^c = E[c \cdot M_1] \quad (2)$$

IV. PRIVATE MATCHING

Freedman et al. [9] presents efficient private matching and set intersection protocol as shown in Fig. 4. The protocol allows two parties to jointly compute the intersection of their inputs without disclosing other information. Supposed that a client (C) and a server (S) want to find their common input as shown, the process starting with

Let $X = \{x_1, \ldots, x_{kc}\}$ be a set of input of C.

Let $Y = \{y_1, \ldots, y_{ks}\}$ be a set of input of S.

The client uses a polynomial to encode his input defined as

$$P(x) = (x-x_1)(x-x_2)\ldots(x-x_{kc}) = a_0 + a_1x + \ldots + a_kx^k \quad (3)$$

The client sends to the server a sequence of ciphertexts

$$E(P(x)) = E(a_0 + a_1 \cdot x^1 + \ldots + a_k \cdot y^k) = E(a_0) \cdot E(a_1)^{x1} \ldots \cdot E(a_k)^{yk} \quad (4)$$

Let r be random number. The server uses homomorphic properties to compute

$$E(r \cdot P(y) + y) \quad (5)$$

The server sends the results back to the client in random order. The client uses his private key to decrypt and gets the elements of intersection $X \cap Y$ without learning other element as shown in Fig.5.

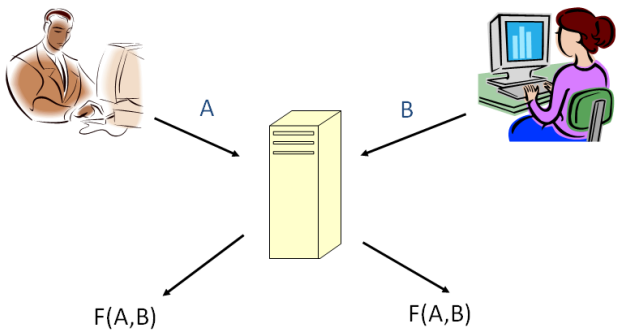

Fig. 4. Private Matching

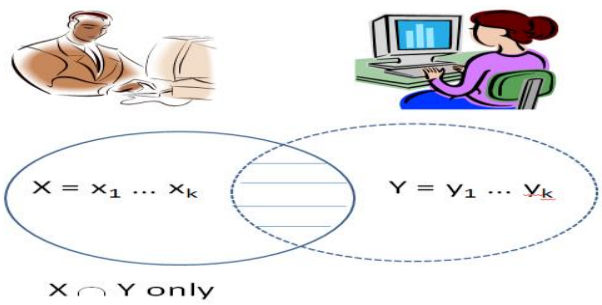

Fig. 5. Evaluation of Private Matching.

V. SYSTEM ARCHITECHTURE

We design the e-learning system as shown in Fig.6. There are three types of client which are student, teacher and admin. Firstly, students and teachers must register for the account. Then, teacher can create lesson and questions using building tools implement with PHP. Student can login and takes a pre-test. After finish taking the test, the student's answer will be encrypted using Paillier cryptosystem and send to the teacher. The teacher compute her answer key with the encrypted message received from the student, shuffle and send back the computed message to the student. The student decrypts the messages and gets "1" for the correct answer and gets random number for the wrong answer. The system can be placed in LTSA system as shown in fig. 7. In the proposed scheme, the learner and coach communicate directly to each other. When the student wants to take a pre-test, the student can directly request questions from the teacher. After finish answering, the student encrypts his set of answer with his public key using Paillier cryptosystem and sends ciphertexts to the teacher. The teacher puts the correct answer in each ciphertexts, shuffle and send back the ciphertexts to the student. Finally, the student decrypts the ciphertexts and gets '1' for the correct answer or gets 'random' for the wrong answer.

VI. PROPOSED SCHEME

Our scheme performs in three phases: encrypting phase, blind matching phase and unbinding phase. The encrypting phase will be performed by the student in order to encrypt his answer and sent the ciphertext to the teacher. The blind matching phase will be performed by the teacher to securely put her answer key into the ciphertext she received and send them back to the student. The unbinding phase will be performed by the student to check for the result of his test by decrypting the received ciphertext. The student has his set of answers. $S= \{s_1,\ldots,s_n\}$ is a set of n answers. For example, the student answers 3 questions then $S = \{s_1, s_2, s_3\}$. The teacher has a set of her answer keys T where $T = \{t_1,\ldots,t_n\}$. The answer s_n corresponding to the key t_n. Let E_s and D_s be public key encryption and decryption for student respectively.

A. Proposed scheme

Our protocol performs as the following steps:

(1) The student generates his own public key, private key and parameters for a semantically-secure homomorphic encryption and decryption scheme.

(2) Encrypting phase:

The student encrypts his answer using his public key and sends all encrypted messages, $E_s(P(x))$ to the teacher.

(3) Blind matching phase:

The teacher picks fresh random number r then evaluates the encrypted polynomial, $E_s(rP(b)+1)$, shuffle and send back to the student.

(4) Unbinding phase;

The student decrypts the received message with his private key wishing have $Ds(E_s(rP(b)+1)) = 1$ which means his answer is correct.

B. Example

Example of our proposed protocol is shown in Table 1. There are 3 questions in a test. The student's answer is a, b and d while the teacher's key is a, c and d respectively. The student begins with encrypting his answers one by one so he gets A1, A2 and A3 then send them to the teacher respectively. The teacher puts the key in the ciphertext using Homomorphic properties and private matching. The teacher does not want the student to learn which question he did correct or incorrect so she shuffles sequence of the messages then sends them back to the student. The student decrypts and gets either 1 for the correct answer or gets random number for the wrong answer. At this point, the student learns only how many points he gets but cannot learn exactly which question he did right or wrong.

VII. CONCLUSION

We have proposed a cryptographical protocol to perform computing score with privacy preserving for the student. Our protocol allows only the student learns his score of the test without disclosing the answer key while the teacher and the server learn.

TABLE I. EXAMPLE OF PROPOSED SCHEME

Student	Teacher
Question 1: a Question 2: b Question 3: d	Question 1: a Question 2: c Question 3: d
A1 = $E_s(P_1(x))$ A2 = $E_s(P_2(x))$ → A3 = $E_s(P_3(x))$	
	B1 = $E_s(rP_1(t_1)+1)$ B2 = $E_s(rP_2(t_2)+1)$ B3 = $E_s(rP_3(t_3)+1)$ ← Shuffle and send B3, B1, B2
Decrypt B3,B1,B2 then get B1 = 1, B2 = random and B3 = 1	

This work is supported by King Mongkut's Institute of Technology Ladkrabang [KREF125609].

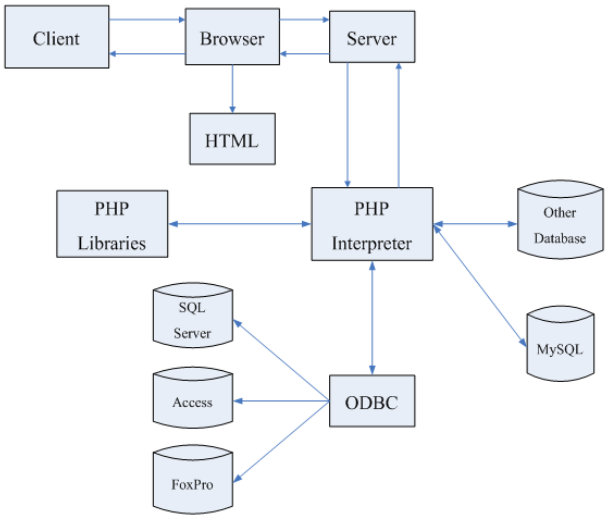

Fig. 6. System Architecture.

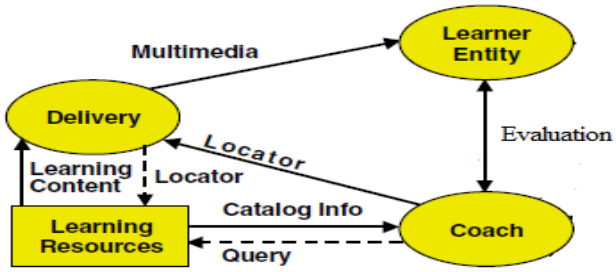

Fig. 7. Out system in LTSA model.

REFERENCES

[1] IEEE standard for learning technology-learning technology systems architecture (LTSA) IEEE Std 1484.1-2003, In IEEE Std 1484, pp. 01-97, 2003.

[2] V. Carchiolo, A. Longheu, M. Malgeri, "Dynamic web pages for tuning formative path", DIIT Technical Report #1013, University of Catania.

[3] Chiu, C., "The Authority Structure Problem of Computer Supported Collaborative Concept Mapping System for Elementary Student", IEEE International Conference on Advanced Learning Technologies (ICALT2001), August 6-8,2001, Madison, WI, USA, Los Alamitos, CA: IEEE Computer Society, 57-56.

[4] Fujiwara, Y., Matsuzawa, B., & Okada.S. , "Learning Assistance Expert System with a Self-Adaptive Function Based on a Causal Network.", The World Multiconference On Systemics, Cybernetics And Informatics, July 22 - 25, 2001, Orlando, Florida, USA.

[5] Thomas, P. and Paine, C, "How Students Learn to Program: Observations of Practical Tasks Completed", Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001), August 6-8,2001, Madison, WI, USA, Los Alamitos, CA, IEEE Computer Society, 170-173.

[6] Kort, B., Reilly, R., & Picard, R. W, "An Affective Model of Interplay between Emotions and Learning: Reengineering Educational Pedagogy – Building a Learning Companion", Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001) , August 6-8,2001, Madison, WI, USA, Los Alamitos, CA, IEEE Computer Society, 43-48.

[7] Abbas, J., Norris, C., and Soloway E, "Analyzing Middle School Students" Proceedings of IEEE International Conference on Advanced Learning Technologies (ICALT2001) , August 6-8,2001, Madison, WI, USA, Los Alamitos, CA: IEEE Computer Society, 107-108.

[8] P. Paillier: "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", EUROCRYPT'99, Springer LNCS, pp. 223-238, May 1999.

[9] M. 1. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection", Eurocrypt 2004, Springer LNCS, pp. 1-19, May 2004

AUTHOR BIOGRAPHY

PERSONAL INFORMATION

NAME	Malanh PHETSAVONG
DATE OF BIRTH	29 April 1990
SEX	Male
NATIONALITY	Lao
PLACE OF BIRTH	Vientiane, Lao PDR

EDUCATION

BACHELOR DEGREE

PROJECT	Budget System of Financial Department, Ministry of Education
FIELD OF STUDY	Information Technology
DURATION	2007-2012
DEPARTMENT	Department of Computer Engineering and Information Technology
FACULTY	Faculty of Engineering
UNIVERSITY	National University of Laos

MASTER DEGREE

THESIS	Privacy-Preserving based Automated Trust Negotiation in E-learning System.
FIELD OF STUDY	Computer Engineering
DURATION	2013-2015
COLLEGE	International College
UNIVERSITY	King Mongkut's Institute of Technology Ladkrabang