Department of Computer Science and Engineering
The Chinese University of Hong Kong

Final Year Project Individual Report (Term 2)

Anonymous Online Course Evaluation

Written by
Yuk Lung Lui

Supervised by
Prof. Lyu Rung Tsong Michael

20 April 2020

Contents

Table of Figures

Abstract

In this semester, our group continue the development of our anonymous online course evaluation system. We implement the system with Hyperledger Fabric and this provides a secure environment for recording our evaluation data. To further improve the security and anonymity of our system, we encrypt the evaluation data with additive ElGamal encryption and conduct the evaluation with offline web application.

In this report, we will focus on the data encryption, system modification and practical evaluation workflow. The previous work in Term 1 and background information will be mentioned in introduction part, which may help readers have better understanding of our work.

In the individual contribution part, I will explain my main contribution to our project. As the encryption part and web UI design are my major tasks and there will be more explanation about these two parts. To know more details about this project, please read the individual report of my partner.

Acknowledgments

We would like to express our gratitude to our supervisor Prof. Michael Lyu and advisor Mr. Edward Yau for providing guidance, feedback and resources. Without their help, this report will not be completed.

We are grateful to everyone who has supported us in this project or read this report.

1 Introduction

1.1 Motivation

Traditional course evaluation is paper based. To complete a course evaluation, we spend lots of time for distributing evaluation forms, filling evaluation forms and collecting evaluation forms. With addition of preparation before the course evaluation and the time for data analysis, the whole course evaluation procedure spends an unreasonable amount of time and resources. To save the cost of evaluation and improve the student experience, digitalize the evaluation process is necessary.

In other countries, some schools already have their online course evaluation systems. This shows that the online course evaluation is feasible. The only problem is getting trust from users. For students, we need to show that their submitted forms are counted and the submitting process would not reveal their identities to teachers. For teachers, we need to show that results are correct, and the results will not be shown to public.

To provide a reliable and trusted online platform, we choose to deploy the system on a blockchain network. Hyperledger Fabric. This blockchain framework promises permissioned membership in the network. This provides a safer and reliable environment to store the evaluation information.

1.2 Term 1 Review

In Term 1, we have setup a basic Hyperledger Fabric network for our online course evaluation system and we could create evaluation, submit evaluation form and calculate evaluation results with our system.

The evaluation system in Term 1 was not complete. Firstly, the web UI was not completed. There was no web UI for creating evaluation and the web UI of other functions is not implemented in same web application. Secondly, the evaluation data was passed without encryption and the data security could be further improved.

In this semester, we have redesigned the web UI design and added encryption method to protect our data. Although the web UI design is not attractive, user should have better user experience with the improved UI. The data encryption is done by additive ElGamal encryption and the we can do the summation with the encrypted data directly. The answer from submitted evaluation data would not be exposed until the tally process. The system architecture is also simplified, and the evaluation workflow will be more efficient.

1.3 Background

1.3.1 Online course evaluation consideration

Allow each student to cast a course evaluation just once

To prevent wrong results because of spamming forms from students or accidents, the evaluation system should prevent multiple forms being sent from a student in the same course. It may be done by some mechanism.

Accurately records the evaluations

The evaluation records should be accurate in anytime. Under cyber-attack, the system should filter the wrong data and prevent the records from being corrupted. If the records are corrupted, there should be a mechanism to restore the data

Accurately counts the evaluations

The system must ensure that the evaluation results are always correct and only the submission from verified students will be counted. Any wrong results should be noticed and removed.

1.3.2 Blockchain

Blockchain is a list of records. Block is the basic storing unit. A block contains a timestamp, hash of current block, hash of the previous block and stored data. These blocks are linked by the hash. This causes the data modification difficult since it will change the hash of block and hash of all successor blocks need to be recalculated.

In addition, the blockchain network may be distributed and sometimes decentralized. Everyone can build a node in the network and access the data. Every change of data or block generation is required approval from the majority nodes. Only when more than half of the network nodes are hacked, the data saved in the blockchain can be changed and it is impossible.

There are two kinds of blockchain, public blockchain and private blockchain.

In public blockchain, anyone without permission can join the network and every node can generate a new block. In most cases, public blockchain uses proof of work mechanism [1] to monitor the block generation and protect data security. If a node wants to generate a new block, the node is required to calculate the nonce, which is a hashed value with a specific feature. Then the node needs to broadcast the new block with the nonce. After the verification from majority of nodes, the new block will be accepted. Calculation of nonce is computationally intensive. Therefore, the block generation is slow and further increase the difficulty of data modification.

In private blockchain, only the trustees can join, and only trusted nodes can generate new blocks. To protect the block generation, some private blockchain use Practical Byzantine Fault Tolerance [1] instead of proof of work. The block generator will be determined by majority. If there are at least two-third honest nodes in the network, this consensus approach will work.

1.3.3 Hyperledger Fabric

Hyperledger Fabric is a permissioned blockchain network framework [2]. As all members know each other, the anonymity of Hyperledger Fabric is weaker than the public blockchain, but it is easier to do user authentication.

When we build a blockchain network, we set up nodes. These nodes are called peers which are the basic units of the network. A peer has its own chaincode and ledger. Chaincode is the code which can access data in the ledger. Ledger records the data by blockchain, and world state is the current value of stored objects. To update the ledger, we need an orderer. It can generate new block and provide ordering service. If we want to share data privately, we can set up channels or create private data collections, which are subsets of organizations share private data in channel.

The workflow of Hyperledger Fabric can be explained as below.

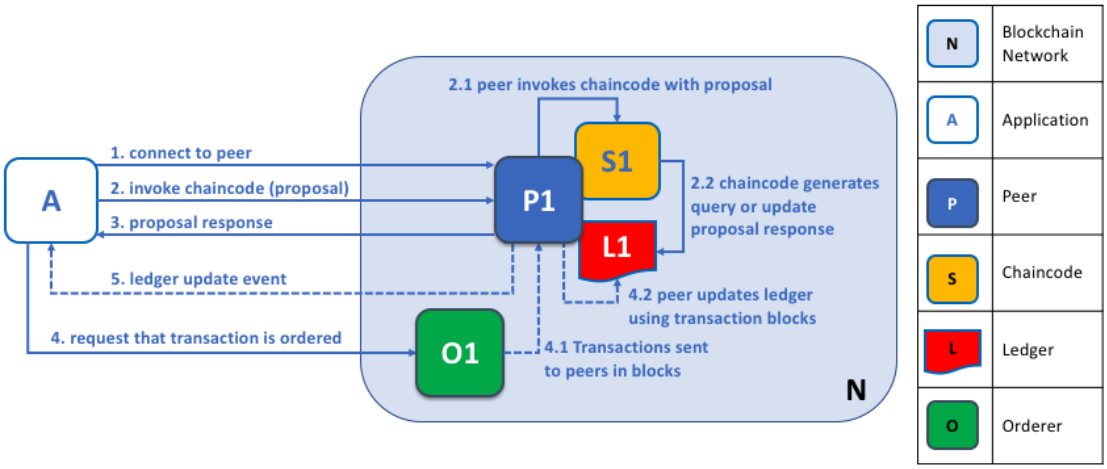


Figure 1: An example of Hyperledger workflow (Source: hyperledger-fabric.readthedocs.io)

Assume we want access the data in ledger, we first invoke the chaincode of the peer through an application. Then the application receives the response from the chaincode. If we just query the data, the requested data will be returned. If we want to update some data, the application will make a transaction and send it to the orderer. Then the orderer generates new blocks. If there are multiple channels in the network and there are multiple update requests at the same time, the orderer will order the requests chronologically. The peer updates its ledger with the blocks. Once the update is completed, the peer will send a notification to the application [3].

1.3.4 Data encryption consideration

Key Security

To encrypt or decrypt data, we need one key for symmetric encryption or a pair of keys of asymmetric encryption. If we use symmetric encryption to encrypt our evaluation data, we must find a way to protect the shared key and share the key among client and server safely. Hybrid encryption would be a possible way. To keep things simple, encrypting data with asymmetric encryption may be a better choice.

Encrypted Data Security

Although data is encrypted, it is still possible to be decrypted by attackers with different methods. To lower the chance of data leakage, the data should be passed with secure network protocol and stored in safe environment. The encryption method should be suitable for the data type and data length.

Homomorphic Encryption

To further increase the data security, the encrypted data should be handled in encrypted form. This can be done by using suitable homomorphic encryption algorithm. With less encryption and decryption processes caused by handling data procedure, the data security should be enhanced. Partial homomorphic encryption should be good enough for our system and it is easier to be realized.

1.4 Objective

The original goal of our project is to build an online anonymous course evaluation system which student can do course evaluation complete anonymously. During the development process, we find that it is too difficult for us to build and prove that our system is truly anonymous. Therefore, the current target for our project is to build an online course evaluation system which teacher or officers cannot track the identity of evaluation participants directly and the system can validate the identity of participants.

In this term, we have these objectives:

1. Study the related research and implementation of data encryption.
2. Improve the workflow of a course evaluation, from preparation to releasing results.
3. Implement the application with complete web UI.

2 Related Study

2.1 Online course evaluation system

When we search “online course evaluation”, we can find out that some colleges use different approaches to do their online course evaluation.

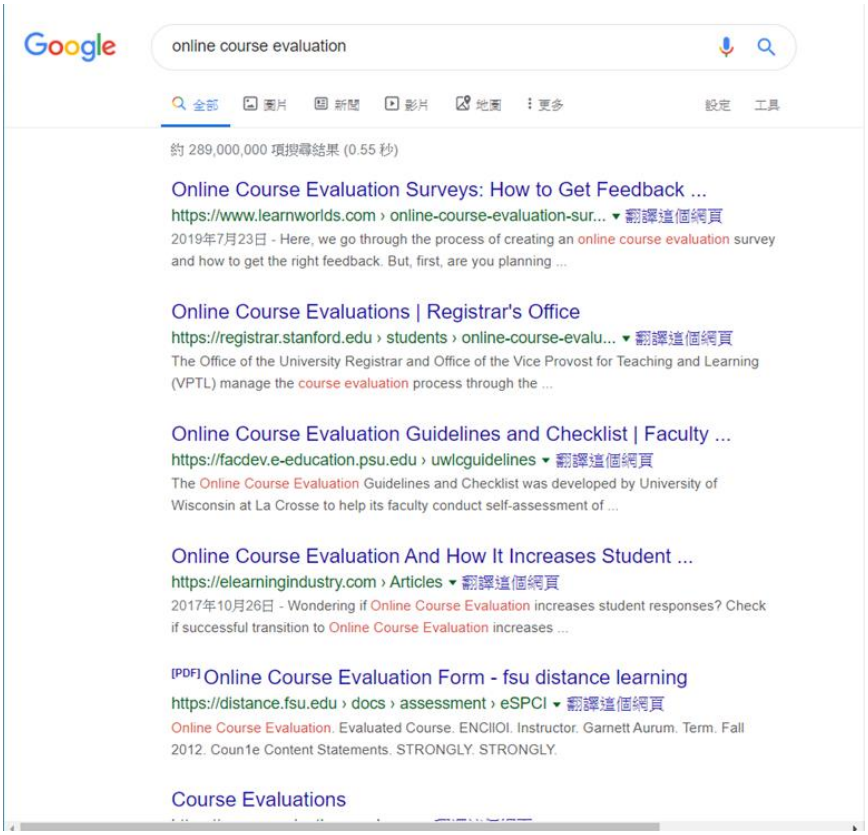


Figure 2: A screenshot of Google search (Source: Google)

Some schools may find outside vendor to manage the evaluation data [4].

Figure 3: A screenshot of Stanford University website (Source: registrar.stanford.edu/students/online-course-evaluations)

However, the websites of those schools never mention how the online course evaluation system works. For the anonymity of the systems, those schools may only mention that the identities of students are confidential without any prove.

2.2 Online e-voting system

As we cannot access the online course evaluation of other schools and understand how those systems work, we try to study the implementations of e-voting system, which are similar with course evaluation system.

2.2.1 Helios

Figure 4: A screenshot of Helios website (Source: https://vote.heliosvoting.org/)

Helios is an open-source e-voting system [5]. To start a vote, we can log in with our Facebook or Google accounts. Then we fill in some setting to generate an election. We can find a group of trustees, or Helios will be the trustee by default. Each trustee holds a unique key pair and the public keys of trustees are used for encrypting the votes.

When we finish our votes, our votes will be encrypted. We will also get ballot

trackers which can be used for vote verifying. There will be a bulletin board which list all submitted ballot trackers. If our trackers are shown on the board, then our vote is counted in the tally.

All encrypted votes are combined into an encrypted tally by homomorphic encryption and only the tally will be decrypted for showing the voting results. To decrypt the tally, all private keys of trustees are needed [6].

2.2.2 Follow My Vote

Figure 5: A screenshot of Follow My Vote website (Source: https://followmyvote.com/)

Follow My Vote is another open-source voting system which is implemented with blockchain [7]. The data of voting is stored in blockchain and the online voting platform uses Elliptic Curve Cryptography (ECC), which is a kind of asymmetric cryptography, to create votes.

Before we start voting, we create two ECC key pairs. Identity key pair is for checking voter identities and voting key pair is for voting. To join a voting, we first show our identities to a verifier and the verifier record our identities with our identity key pair. Then we register our voting key pair with one of the identity keys anonymously. Then we can make a vote and sign the vote with our private voting key. Every voter can use the public voting key to verify our vote [8].

2.2.3 Polys

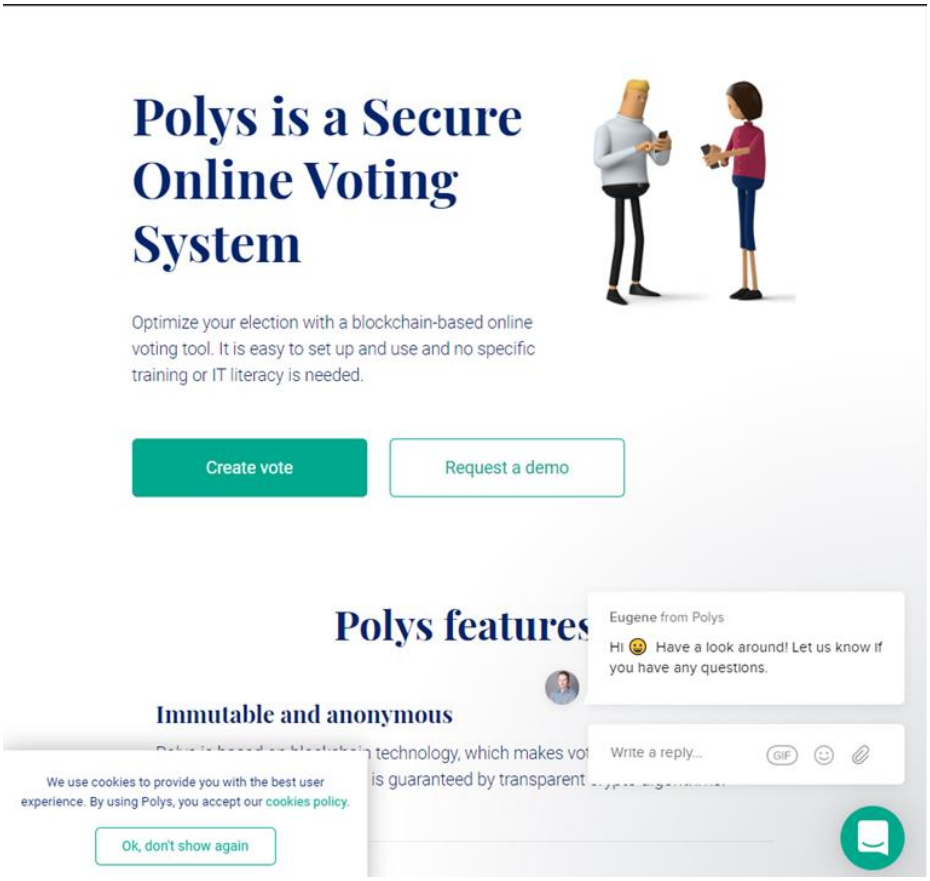


Figure 6: A screenshot of Polys website (Source: https://polys.me/)

Polys is a blockchain-based online voting platform [9]. In a voting, each voter has a token and a random Ethereum account. To vote, the voter need to submit the token to registry. The registry will find an alias and the registry will return the address of alias to the voter. Then the voter can cast a vote with the alias. The alias will check the Ethereum account. If the Ethereum account is verified, the alias will cast the vote [10].

2.3 Encryption Algorithm

To protect the data and privacy of user, the data will be transferred in encrypted form and the encrypted data will be directly used in tally process. The additive homomorphic cryptosystem is suitable for our system.

2.3.1 Benaloh cryptosystem

Benaloh cryptosystem is homomorphic additive public key cryptosystem, which is created by Josh Benaloh [11]. Below is the procedure of using Benaloh cryptosystem.

1. **Key generation:**

   For block size r:

   We need 2 big primes, P and Q.

   P and Q must satisfy below conditions:

$$r|(P-1), \gcd\left(r, \frac{P-1}{r}\right) = 1$$

$$\gcd\big(r, (Q-1)\big) = 1$$

   Then we need N, B, Y:

$$N = PQ$$

$$B = (P-1)(Q-1)$$

$$For\ r = P_1 P_2 \dots P_k, Y \in \mathbb{Z}_N^*, For\ each\ P_i, Y^{\frac{B}{P_i}} \not\equiv 1\ mod\ N$$

   Finally, we need X:

$$X = Y^{B/r} mod\ N$$

Up to here, we have public key (Y, N) and private key (B, X).

2. **Encryption:**

   To encrypt message m, which $m \in \mathbb{Z}_r$:

   Pick a random value u, which $u \in \mathbb{Z}_N^*$.

   Encrypt m.

$$E(m) = Y^m u^r\ mod\ N$$

3. **Decryption:**

   First compute $a = E(m)^{B/r}\ mod\ N$.

   Find m which $m = \log_X a$.

If we have the private key, we find m easily by simple exhaustive search or other discrete logarithm computing algorithms. We can add two encrypted messages by $E(m1) \cdot E(m2)$ and the decrypted sum will be $m1 + m2$.

2.3.2 Paillier cryptosystem

Paillier cryptosystem is another homomorphic additive public key cryptosystem, which is invented by Pascal Paillier [12]. Below is the procedure of using Paillier cryptosystem.

1. **Key generation (Here a simpler method is presented):**

   We need P and Q, 2 random and independent primes with equal length.

   Then we need N, G, Y, X:

$$N = PQ$$

$$G = N + 1$$

$$Y = (P-1)(Q-1)$$

$$X = \big((P-1)(Q-1)\big)^{-1} \bmod N$$

   Up to here, we have public key (N, G) and private key (Y, X).

2. **Encryption**

   M is the message to be encrypted where $0 \leq M \leq N$.

   Then we need random r where $0 < r < N,\ \gcd(r, N) = 1.$

   Now we can encrypt M:

$$E(M) = G^M r^N \ mod\ N^2$$

3. **Decryption**

   Decrypt encrypted M:

$$M = \left(\frac{(E(M)^Y \ mod\ N^2) - 1}{N}\right) \cdot X \ mod\ N$$

   Just like many additive homomorphic cryptosystems, we can add the encrypted messages by $E(m1) \cdot E(m2)$ and the decrypted sum will be $m1 + m2$.

2.3.3 Additive ElGamal cryptosystem

ElGamal cryptosystem is homomorphic multiplicative public key cryptosystem, which is invented by Taher Elgamal [13]. This cryptosystem can be homomorphic additive with a modification. Below is the procedure of using Additive ElGamal cryptosystem.

1. **Key generation:**

   We need a very large prime P.

   From $(1, \dots, P-1)$, we choose G and X where $\gcd(G, X) = 1$.

   Then we need Y:

$$Y = G^X \ mod \ P$$

   Up to here, we have public key (Y, G, P) and private key X.

2. **Encryption**

   M is the message to be encrypted.

   Then we need random r where $1 \leq r < P - 1, \ \gcd(r, P) = 1$.

   Now we can encrypt M:

$$E(M) = (G^r mod \ P, Y^r G^M \ mod \ P)$$

3. **Decryption**

   Decrypt encrypted M:

$$G^M = ((((G^r mod\ P)^X mod\ P)\ modInv\ P) \cdot (Y^r G^M\ mod\ P))mod\ P$$

As G and P is known, we can get back M by simple exhaustive search or other discrete logarithm computing algorithms.

Just like many additive homomorphic cryptosystems, we can add the encrypted messages by $E(m1) \cdot E(m2)$ and the decrypted sum will be $m1 + m2$.

3 Design

3.1 Overview

Hyperledger Fabric is used as our system network framework. The major reason is that Hyperledger Fabric is a permissioned blockchain network framework and people cannot access it without permission. If we transfer the data through channels and private data collections (PDC), the data only pass to the related parties. With these properties, Hyperledger Fabric provides a reliable environment to store data.

The Term 2 system design is different from what we proposed in Term 1. The Term 2 design is simpler than Term 1 design as we have removed the key signature part and we use token to validate the identities of participants directly.

3.2 Principle of system

3.2.1 User validation by token

Before starting a course evaluation, it is important to ensure that only the students who study the course can join the course evaluation. Therefore, the students should receive a token from the system. This token is generated by our evaluation system. To join the evaluation, the system will verify their identities with their tokens.

In our Term 2 design, the identities of participant are still validated by token. However, the record of tokens is different. For our Term 1 design, there are records of tokens and their owners. To further reduce the risk of token information leakage, tokens will be distributed by our system without leaving owner records.

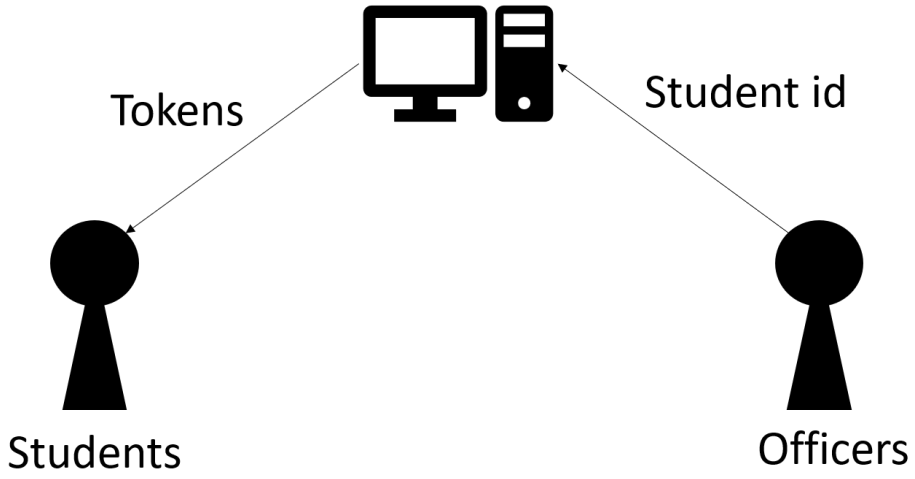


Figure 7 New Token generation and distribution

3.2.2 Evaluation data verification

For our Term 2 design, we have removed the RSA signature step and use ElGamal Encryption to protect and verify the evaluation result data.

In our old design, the evaluation data is verified by RSA signature. This can ensure the evaluation data is submitted by the people who hold the private key of registered RSA public key. We try to revise this approach and find that the data verification can be done in simpler way with data encryption and new token generating mechanism.

Stage 1: Evaluation submission

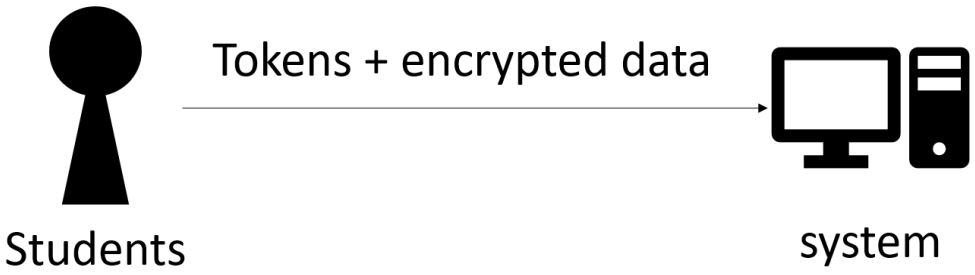


Figure 8 Evaluation submission

When the students get the token from the system, they can join the evaluation with their tokens. Once they finish the evaluation, the client-side application encrypts the data with ElGamal public key and send the ciphertext with the tokens to server side. The token is used for verification.

Stage 2: Handling Evaluation data

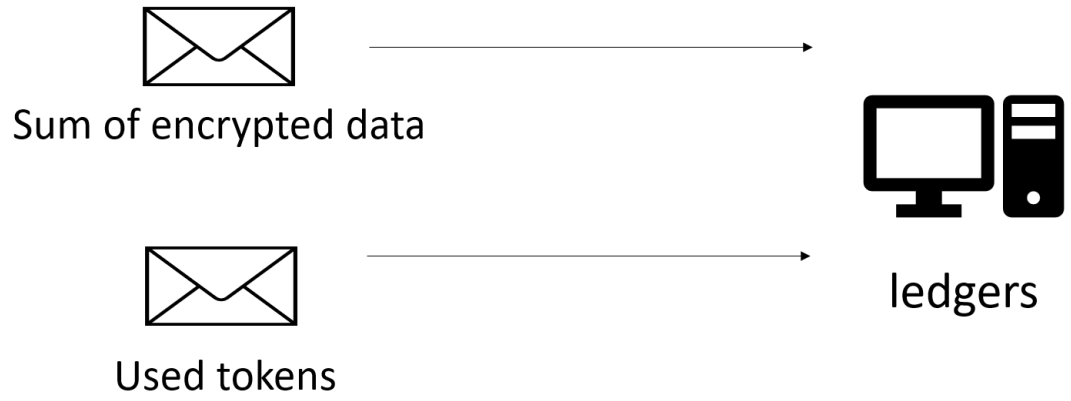


Figure 9 Update ledgers with verified token

When the server-side receives the tokens with encrypted data, the token would be checked. If the token can be found in record and never used, the received encrypted data would be added to the saved encrypted data. The token would be marked as used. After updating the saved encrypted data and used token, the data modification would be recorded in ledgers.

Stage 3: Check the results

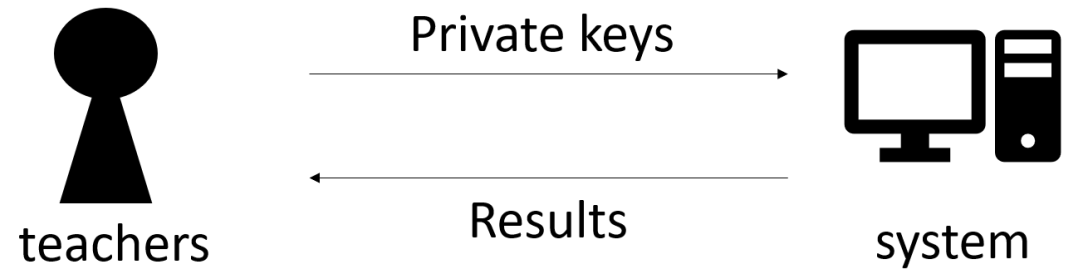


Figure 10 Receive and check the results

After the deadline of submitting evaluation, the teachers can submit their private key to the system. The system uses the private key to decrypt the evaluation results and return the results to teachers. The results include the total number of used tokens. If the number of used tokens match the evaluation results, the evaluation results should be valid.

The improvement made by new approach

As we use token to do validation, the workflow of whole system is simplified and more efficient. We can skip the RSA key generation, RSA public key recording, RSA signature verification.

For the data security, as the server-side never stored the information of ElGamal private key, the evaluation data is always in encrypted format. This enhances the data security.

However, there is a small issue when we choose this new approach. Students cannot check their submitted evaluation data easily. If the students want to check their submission, they can only receive the encrypted unreadable data. It is not a big problem for course evaluation because students should not be able to check their submission, just like the current course evaluation.

3.3 Completed workflow

3.3.1 Create new evaluation

The initiator provides issuer identity, course ID, list of student id and evaluation expiration date to our web application. The application sends create evaluation request to the server-side. The server validates the issuer identity. After passing the validation, a new evaluation would be created with given information and random generated tokens will be distributed to the students according to the given student id list. An ElGamal key pair would be generated and the system keeps the public while the private key would be sent to the initiator.

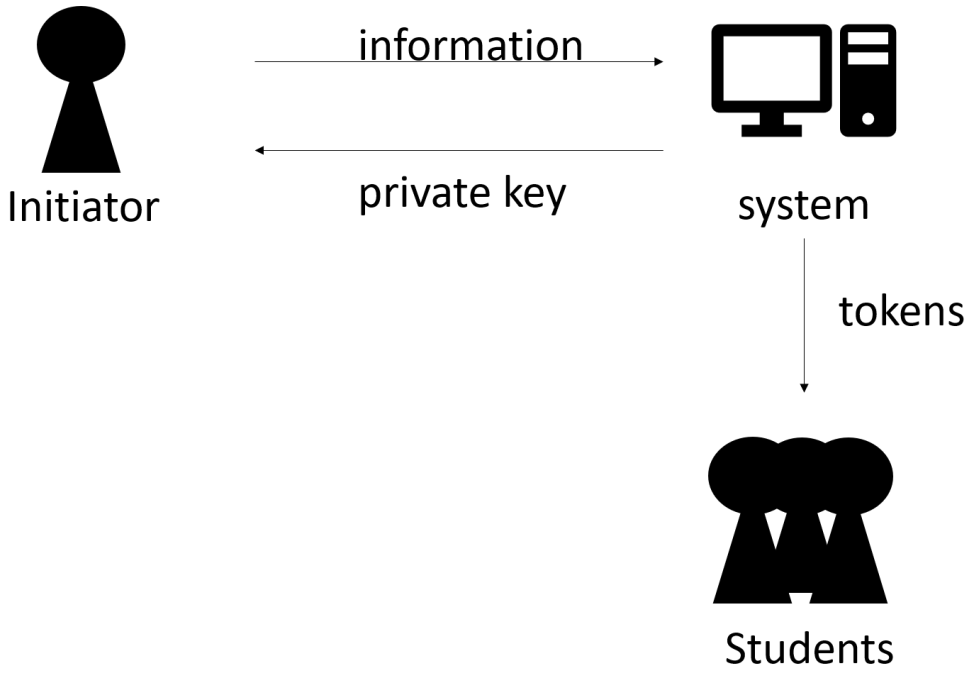


Figure 11 Create new evaluation

3.3.2 Submit evaluation

Students can join the evaluation and submit their finished evaluation form with their tokens. The evaluation data would be encrypted by the ElGamal public key before the submission.

Once the server-side receives the submission, the related saved encrypted data and token record would be fetched from the ledgers. The new received token would be checked. If the token is valid, the token record would be updated and the new received encrypted evaluation data would be added to the old encrypted data. Then the updated data would be saved in the ledgers.

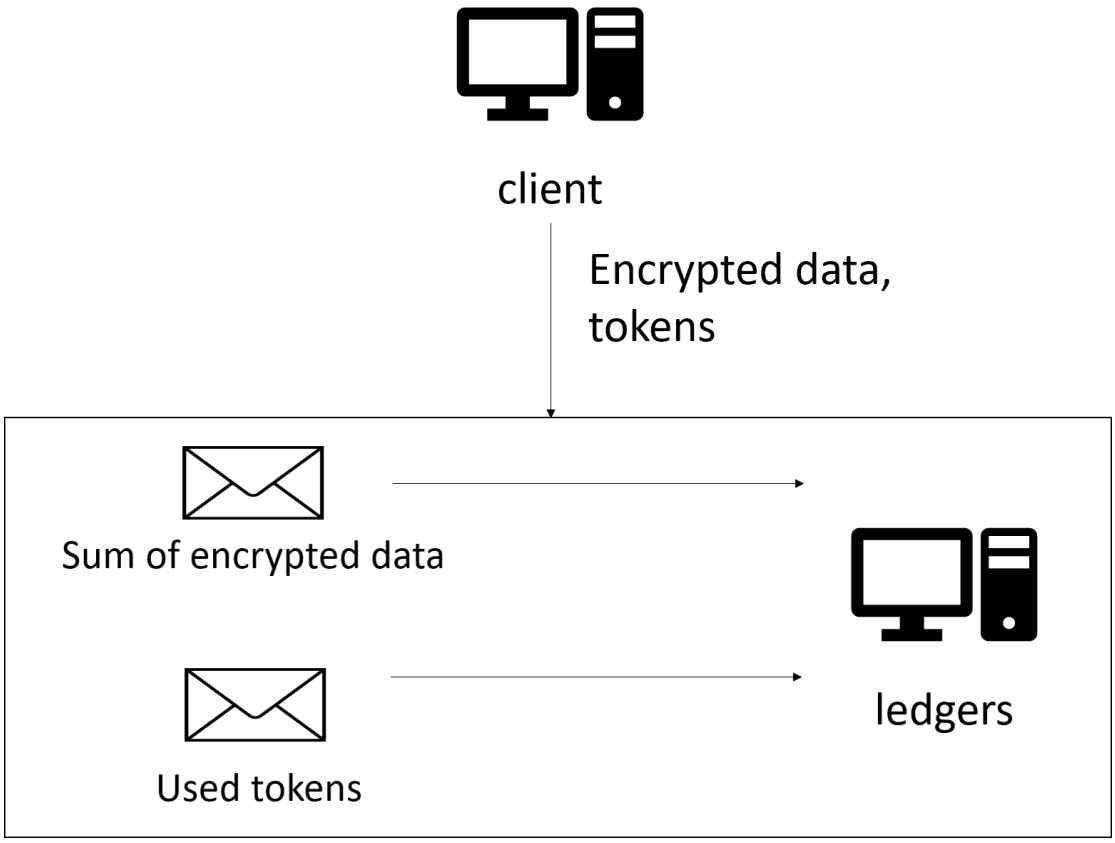


Figure 12 Submit evaluation

3.3.3 Viewing results

The student can submit their evaluation form until the expiration date. After the expiration date, the private key owner can view the results by submit the private key. The encrypted evaluation data and number of used tokens would be returned, and the data would be decrypted locally. After finishing decryption, the result would be shown. The owner can check the results.

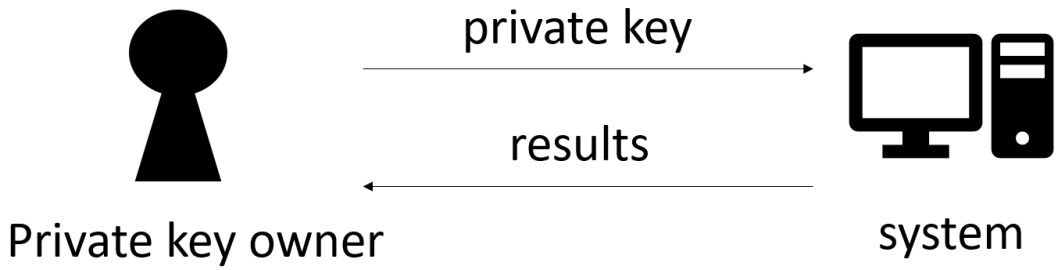


Figure 13 Viewing results

3.4 Expectation for proposed design

For our current design, the tokens are only known by the students and system. The private keys are kept by the evaluation initiators. Teachers can only view the results. Officers can access the server, but they cannot decrypt the encrypted data and they do not know the token owners. Students can only submit evaluation data. Although it is not a truly anonymous system based on zero knowledge proof, we expect that this can protect the privacy of both teachers and students.

3.5 Assumptions

We have made some assumptions when we design:

3.5.1 Honest party who hold the system

This assumption is the most important assumption and cannot be failed.

If the parties are not trustworthy, they can do any modification to the system and the system cannot be worked as expected.

There is no solution for this problem. Fortunately, this assumption is difficult to fail as there is no reason for faculty officers doing any harmful modification.

3.5.2 The token and private key are distributed correctly

In current design, the data security is based on the tokens and private key. The tokens and private key must be distributed to correct target. Otherwise, the results of evaluation and the user privacy will be affected.

3.5.3 Stable network connection

We assume that all the instructions are run on stable network connection. If the system is run with an unstable network, the data transfer may be affected and causing unpredictable error.

3.5.4 User has secure mailbox

We assume that the mailboxes of system users are secure, and no one can get the email information with illegal way. If the mailbox is not safe, the tokens which sent by email may be known by others.

4 Implementation

4.1 Overview of 2nd term implementation

In the 2nd term, we implement an online evaluation system with Hyperledger Fabric. The system contains two parts: the blockchain network and the client-side web interfaces.

Hyperledger Fabric handles all the data transactions and data storing. Adding tokens, verifying tokens, adding evaluation results and querying evaluation results, all of these features are implemented by chaincodes to ensure the integrity and reliability of the transactions.

For better user experience, the client-side web interfaces are necessary and normal users can interact with the online course evaluation system easily.

4.2 Blockchain network

4.2.1 Framework and programming language

Hyperledger Fabric is a well-known open-source permissioned distributed ledger technology platform [14]. It provides Node.js SDK for chaincode [15] and application (client) [16]. For easier development, JavaScript is chosen to be our main programming language. We can use JavaScript across the whole development cycle. This helps us integrate different parts of the system.

For the database part, CouchDB [17] is used. CouchDB is an open source document-oriented NoSQL database. It stored data in JSON format and JavaScript is its query language. As it is a document-oriented NoSQL database, we can save different kinds of document easily. It is more convenient when we can query the data with JavaScript.

Node.js is an open source, cross platform JavaScript runtime environment. With Node.js, we can run JavaScript on server-side and use different Node packages to help our development.

4.2.2 System architecture

Here is a diagram of the 1st term blockchain network system architecture:

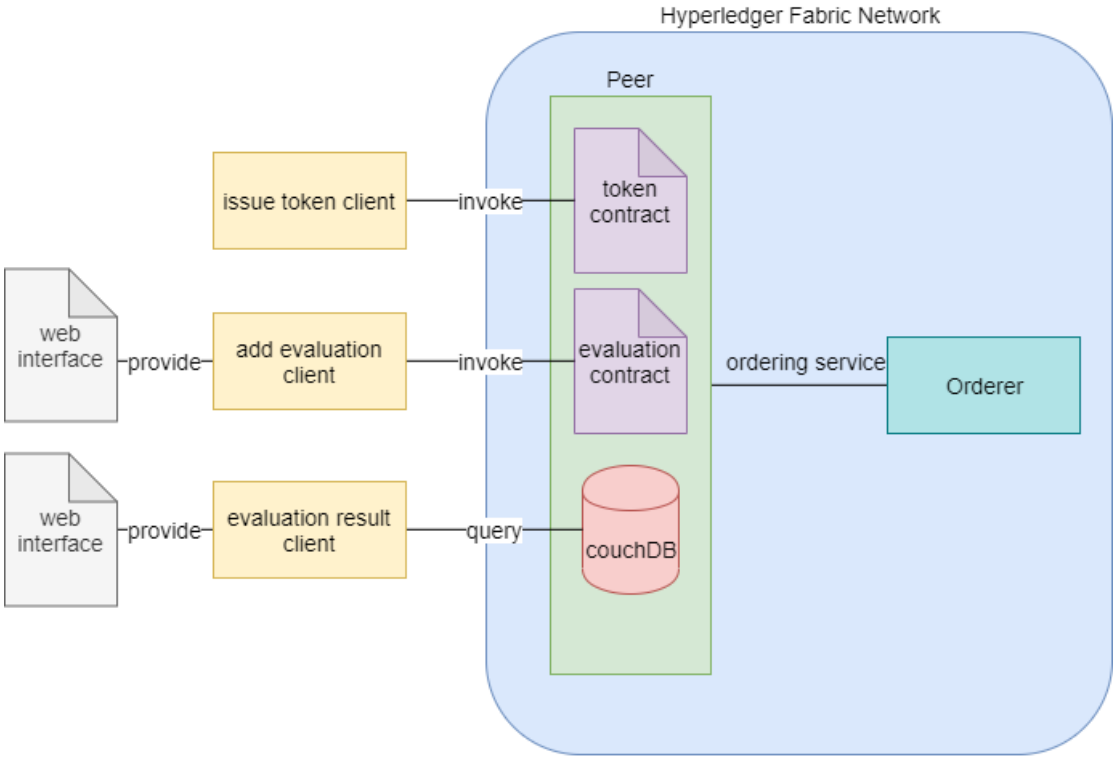


Figure 14: System architecture diagram of blockchain network [18]

Peer

The peer in our current Hyperledger Fabric network handles token generation, evaluation verification and data storage.

When we send a request from the issue token client, the peer invokes the token contract. After the updating information is received by orderer, new token record block will be added in the ledger. The token record contains the token id, token status (used or not), the expiration date and course id.

If students submit an evaluation form, the peer invokes the evaluation

contract and verifies the request by checking the provided token. If the token exists and it is never used before, orderer will receive the updating information and add new blocks. Otherwise, the evaluation form will be refused, and a refuse response will also be sent to the client-side.

To view the evaluation results, we can make a request through the evaluation result client. The peer will provide the results if the request is valid.

Orderer

The orderer is responsible for the block generation and ordering. Once the client request is verified, the client can send the information generated from the response of peer, the orderer will generate the block which stores the provided information.

Chaincode (smart contract)

Token contract is the chaincode which is responsible for querying and updating token information in the ledger. This chaincode checks the client request and give different responses to the client.

Evaluation contract is the chaincode which is responsible for querying and updating evaluation information in the ledger.

Database

CouchDB will save all the data, including the token information and evaluation results.

Client

In current implementation, issue token client, add evaluation client and evaluation result client are implemented in a single web application.

In issue token client, we can make requests of issuing new tokens for course evaluation. By giving the information of issuer, course ID, students id and token expiration date, the client can request the system to generate new tokens with the given information. Once the request is accepted, the tokens will be generated and distribute to the students email directly.

We can submit our evaluation form through add evaluation client. Once we finish evaluation, we can submit the form and the client will make a request of adding evaluation. If the form is not finished or the provided tokens are not valid, there will be warning to remind the users.

The results of course evaluation can be viewed in evaluation result client. The user input a private key of the related course. The client will send a checking request to the server and the server will return a random value and its ciphertext. If the input private key can decrypt and give the correct value, the client will send request and the encrypted evaluation results will be returned. The results will be decrypted locally and then user view the target course evaluation results.

4.3 Web interface

4.3.1 Framework and programming language

In this term, we continue to use React with React-Bootstrap as our web interface framework. As we would like to build a single web application, React would be a suitable framework. In the following section, some screenshots of the main pages will be shown.

4.3.2 User Interface

4.3.2.1 Sign in

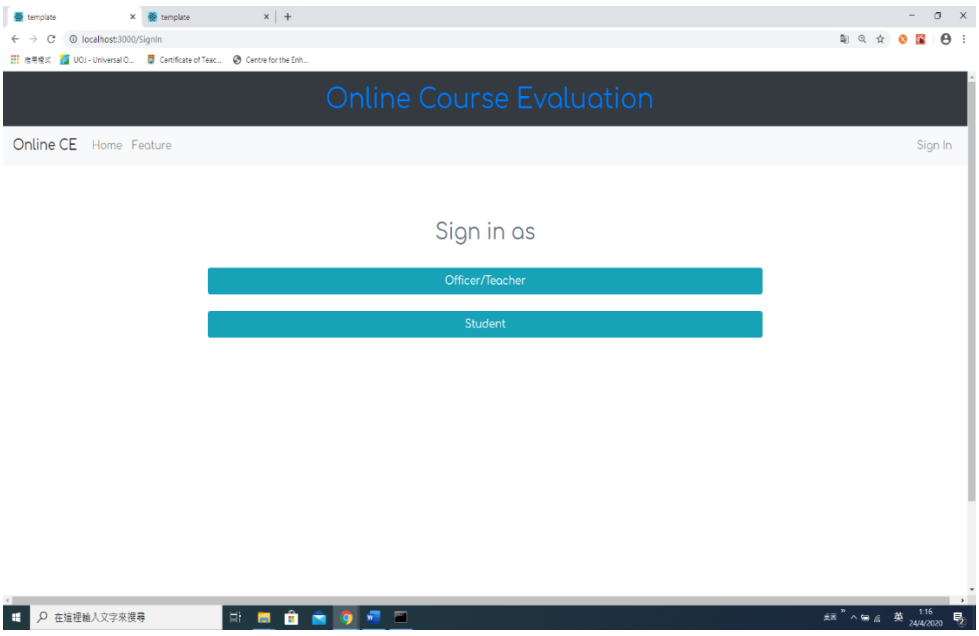


Figure 15 Sign in page

In the sign in page, there is two buttons. When users sign in as teachers or officers, the user should click “Officer/Teacher” button and the login popup will appear. If user is student, the user should click “Student” button and join the evaluation.

4.3.2.2 Officer Page

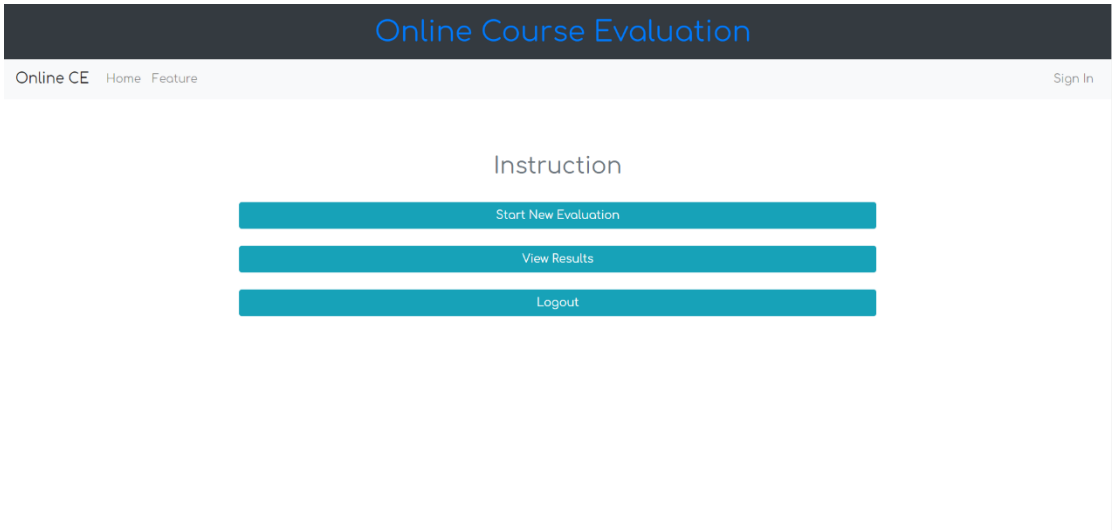


Figure 16 Officer page

For the users who are officer or teachers, this is the homepage after they login the system. Click the “Start New Evaluation” to create new evaluation. Click “View Results” will redirect user to the result page to check the evaluation result. Click “Logout” will redirect user to sign in page.

4.3.2.3 Result page

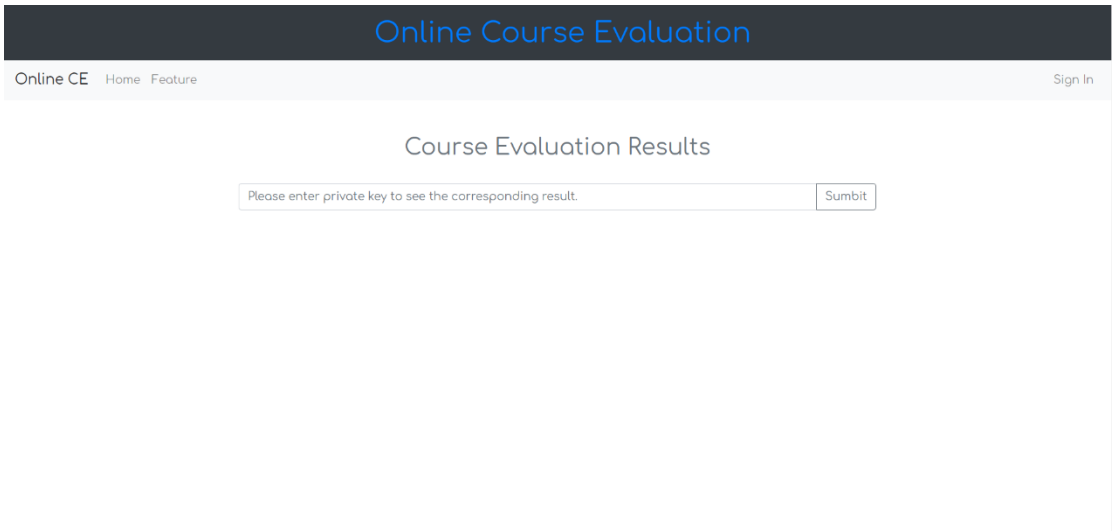
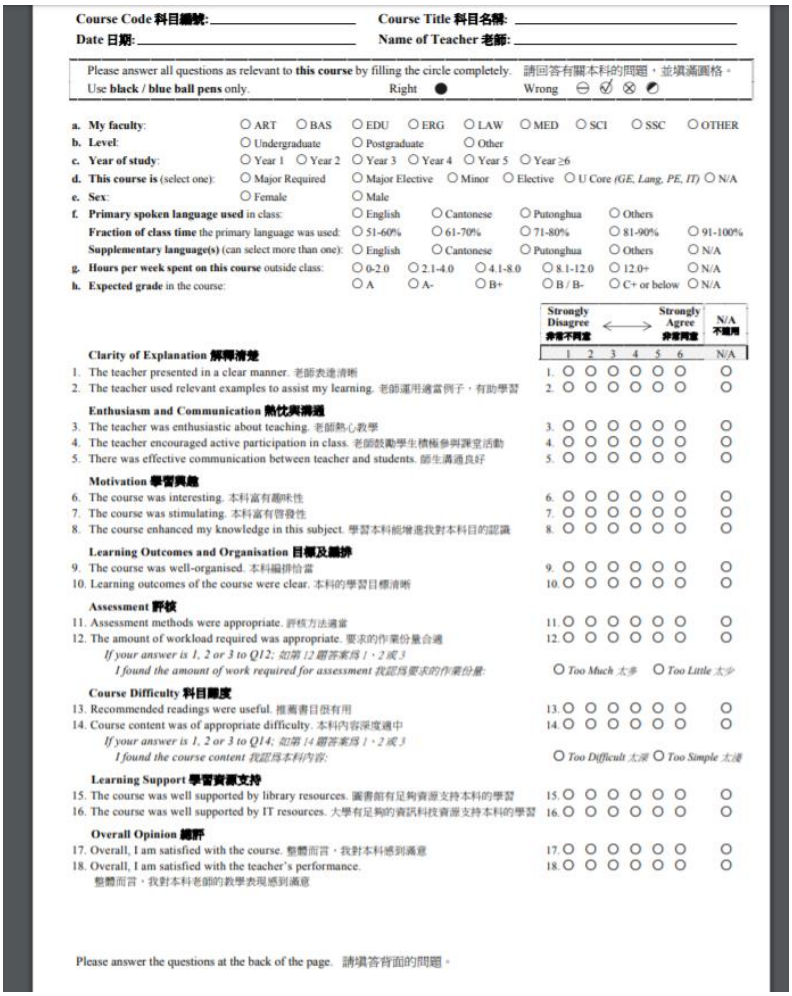


Figure 17 Result page

Only officer or teachers can access the result page. To check the evaluation results of a course, user should enter the private key of that course evaluation into

the provided input box. If the private key is correct, the results will be shown below the input box and the results will be presented in table form.

4.3.2.4 Evaluation form page



Course Code 科目編號: _____ Course Title 科目名稱: _____

Date 日期: _____ Name of Teacher 老師: _____

Please answer all questions as relevant to **this course** by filling the circle completely. 請回答有關本科的問題，並填滿圓格。
Use **black / blue ball pens** only. Right ● Wrong ⊖ ✓ ⊗ ◐

a. **My faculty**: ○ ART ○ BAS ○ EDU ○ ERG ○ LAW ○ MED ○ SCI ○ SSC ○ OTHER
b. **Level**: ○ Undergraduate ○ Postgraduate ○ Other
c. **Year of study**: ○ Year 1 ○ Year 2 ○ Year 3 ○ Year 4 ○ Year 5 ○ Year ≥6
d. **This course is** (select one): ○ Major Required ○ Major Elective ○ Minor ○ Elective ○ U Core *(GE, Lang, PE, IT)* ○ N/A
e. **Sex**: ○ Female ○ Male
f. **Primary spoken language used** in class: ○ English ○ Cantonese ○ Putonghua ○ Others
Fraction of class time the primary language was used: ○ 51-60% ○ 61-70% ○ 71-80% ○ 81-90% ○ 91-100%
Supplementary language(s) (can select more than one): ○ English ○ Cantonese ○ Putonghua ○ Others ○ N/A
g. **Hours per week spent on this course** outside class: ○ 0-2.0 ○ 2.1-4.0 ○ 4.1-8.0 ○ 8.1-12.0 ○ 12.0+ ○ N/A
h. **Expected grade** in the course: ○ A ○ A- ○ B+ ○ B / B- ○ C+ or below ○ N/A

	Strongly Disagree 非常不同意 ⟷ Strongly Agree 非常同意						N/A 不適用
	1	2	3	4	5	6	N/A
Clarity of Explanation 解釋清楚							
1. The teacher presented in a clear manner. 老師表達清晰	○	○	○	○	○	○	○
2. The teacher used relevant examples to assist my learning. 老師運用適當例子，有助學習	○	○	○	○	○	○	○
Enthusiasm and Communication 熱忱與溝通							
3. The teacher was enthusiastic about teaching. 老師熱心教學	○	○	○	○	○	○	○
4. The teacher encouraged active participation in class. 老師鼓勵學生積極參與課堂活動	○	○	○	○	○	○	○
5. There was effective communication between teacher and students. 師生溝通良好	○	○	○	○	○	○	○
Motivation 學習興趣							
6. The course was interesting. 本科富有趣味性	○	○	○	○	○	○	○
7. The course was stimulating. 本科富有啓發性	○	○	○	○	○	○	○
8. The course enhanced my knowledge in this subject. 學習本科能增進我對本科目的認識	○	○	○	○	○	○	○
Learning Outcomes and Organisation 目標及組排							
9. The course was well-organised. 本科編排恰當	○	○	○	○	○	○	○
10. Learning outcomes of the course were clear. 本科的學習目標清晰	○	○	○	○	○	○	○
Assessment 評核							
11. Assessment methods were appropriate. 評核方法適當	○	○	○	○	○	○	○
12. The amount of workload required was appropriate. 要求的作業份量合適	○	○	○	○	○	○	○
If your answer is 1, 2 or 3 to Q12; 如第 12 題答案爲 1、2 或 3							
I found the amount of work required for assessment 我認爲要求的作業份量:	○ *Too Much 太多*		○ *Too Little 太少*				
Course Difficulty 科目難度							
13. Recommended readings were useful. 推薦書目很有用	○	○	○	○	○	○	○
14. Course content was of appropriate difficulty. 本科內容深度適中	○	○	○	○	○	○	○
If your answer is 1, 2 or 3 to Q14; 如第 14 題答案爲 1、2 或 3							
I found the course content 我認爲本科內容:	○ *Too Difficult 太深*		○ *Too Simple 太淺*				
Learning Support 學習資源支持							
15. The course was well supported by library resources. 圖書館有足夠資源支持本科的學習	○	○	○	○	○	○	○
16. The course was well supported by IT resources. 大學有足夠的資訊科技資源支持本科的學習	○	○	○	○	○	○	○
Overall Opinion 總評							
17. Overall, I am satisfied with the course. 整體而言，我對本科感到滿意	○	○	○	○	○	○	○
18. Overall, I am satisfied with the teacher's performance. 整體而言，我對本科老師的教學表現感到滿意	○	○	○	○	○	○	○

Please answer the questions at the back of the page. 請填答背面的問題。

Figure 18 Original course evaluation form multiple choices questions (*Source: https://www.cuhk.edu.hk/clear/qm/A7-1.pdf)*

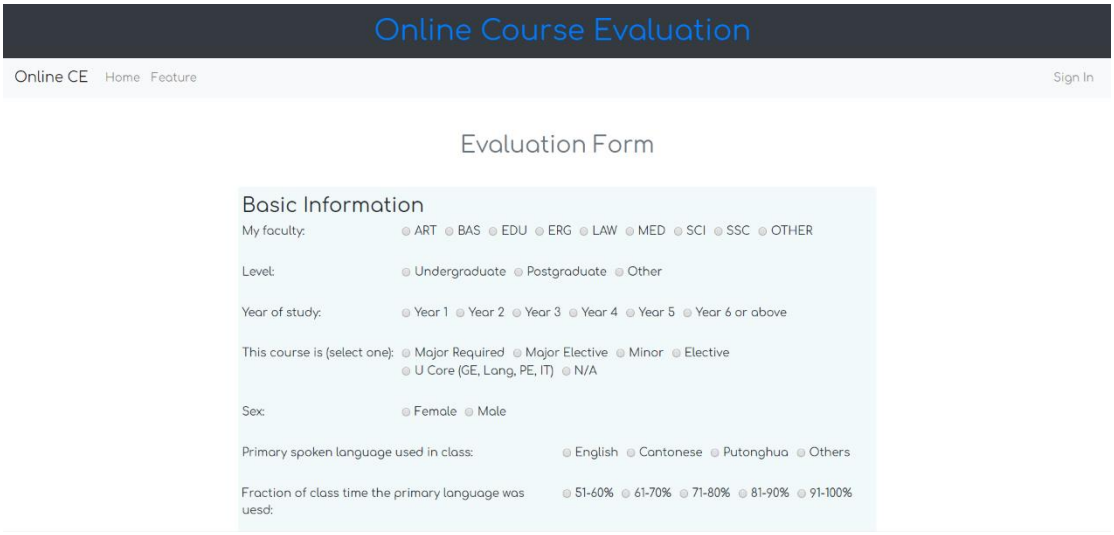


Figure 19 Evaluation form page

The evaluation form page can be accessed by anyone. The evaluation form shown in the page has all the multiple choices questions of the original paper-based course evaluation form. Students must finish the form before submission. Although everyone can access this page, only student with valid token can successfully submit the evaluation form. When we submit the form, the form will be sent as JSON format as below:

```
{
“Q1”:[“0”, “0”, “0”, “1”, “0”, “0”, “0”],
“Q2”:[“0”, “0”, “0”, “0”, “1”, “0”, “0”],
“Q3”:[“0”, “0”, “0”, “0”, “0”, “1”, “0”],
                    .
                    .
                    .
“Q18”:[“0”, “0”, “1”, “0”, “0”, “0”, “0”]
}
```

Figure 20 Data format Example

4.4 Demonstration

The demo below is just a showcase of workflow. Screenshots are not taken from real testing.

4.4.1 Stage 1: Create a new evaluation

To create a new evaluation, we first need to sign in with an officer/ teacher account. The username and id of the officer/ teacher account will be given to corresponding staff directly and no sign-up procedure is required.

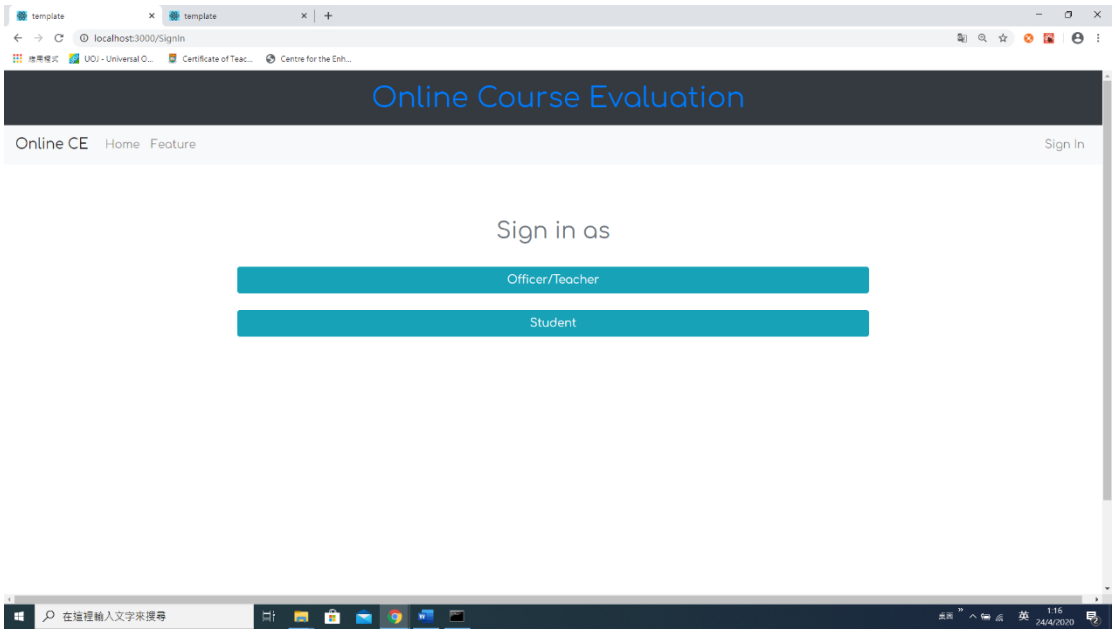


Figure 21 Go to the sign in page

Click the “Officer/Teacher”, and a login popup will appear.

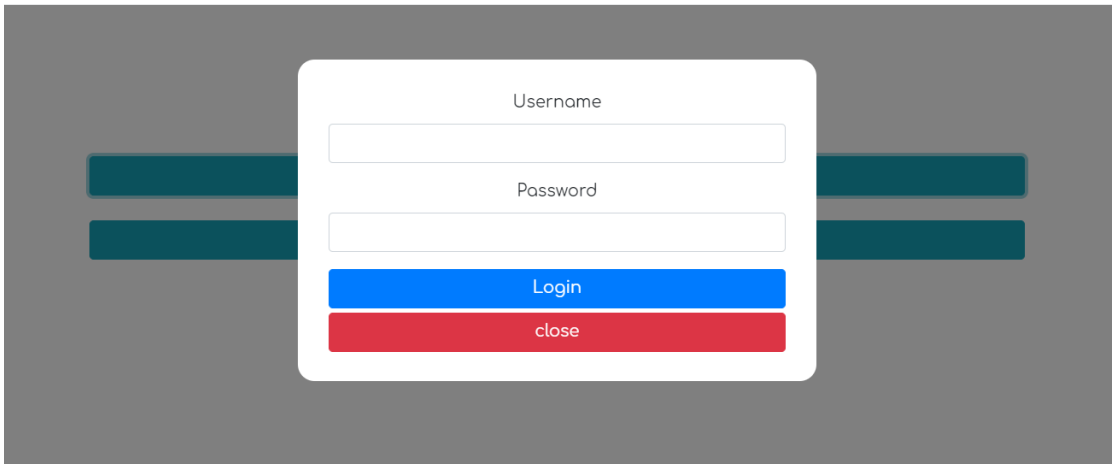


Figure 22 Login popup

Login with the correct username and password, and we can reach the Officer page.

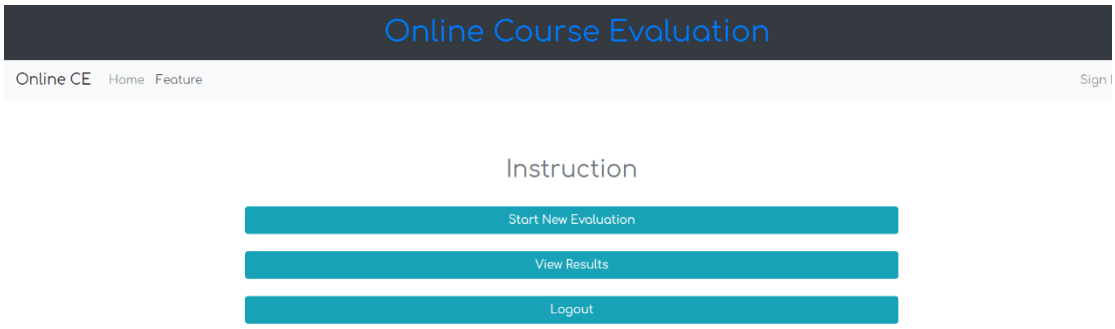


Figure 23 Successful Login

Click “Start New Evaluation”, and a popup will appear.

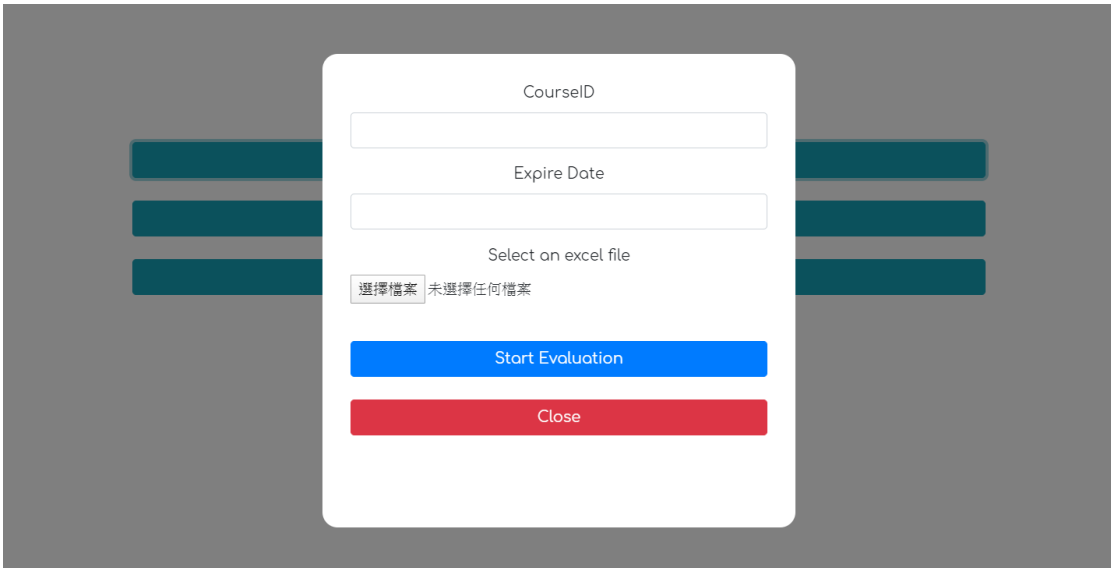
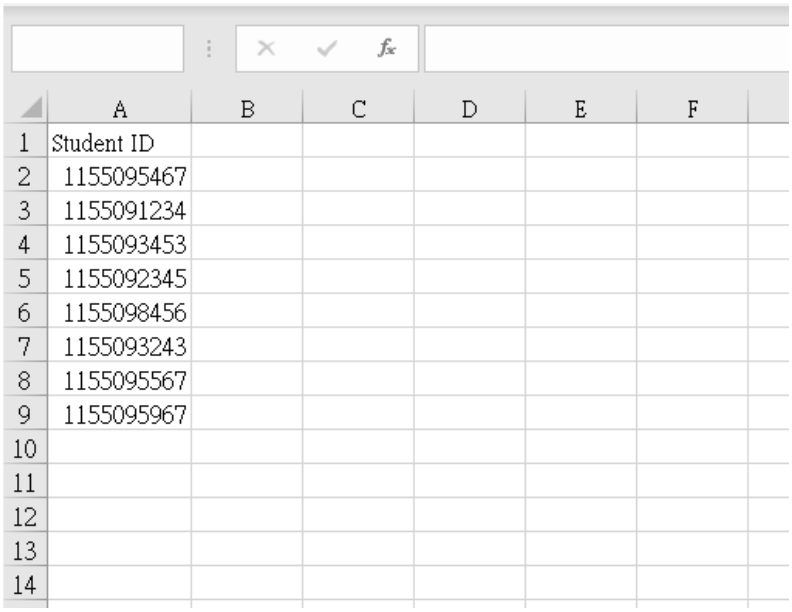


Figure 24 Evaluation popup

Enter the required information. For the file input, please select an excel file with this input format.



	A	B	C	D	E	F
1	Student ID					
2	1155095467					
3	1155091234					
4	1155093453					
5	1155092345					
6	1155098456					
7	1155093243					
8	1155095567					
9	1155095967					
10						
11						
12						
13						
14						

Figure 25 Excel file input format

After entering all required information, click “Start Evaluation” and a new

evaluation will be created. The ElGamal key pair will be generated locally. The public key will be sent to server and the private key will be shown in a popup window. The private key should be kept well.

4.4.2 Stage 2: Submit an evaluation form

After the evaluation is created, the students should receive an email in their CUHK link mailboxes. The email contains a link to evaluation form page. Click into the link.

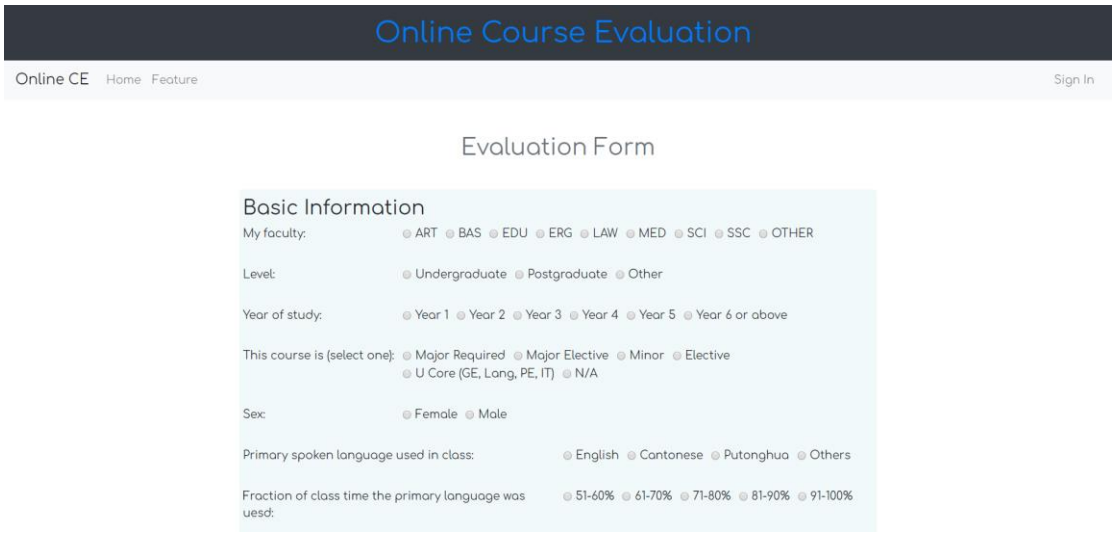


Figure 26 Redirect to the evaluation form page

Finish the form and click submit. The page will redirect to the submit page.

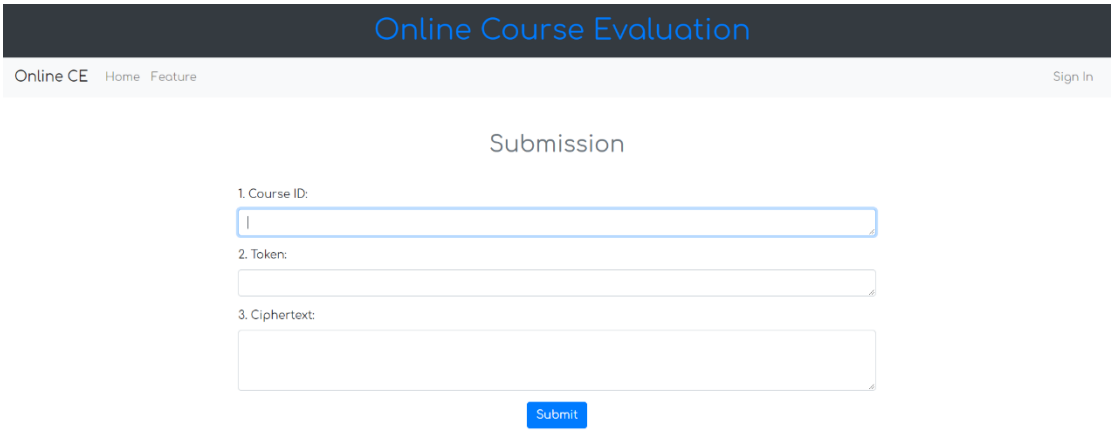


Figure 27 Submission page

The textbox will be automatically filled in actual case. The student just need to click “Submit” button.

4.4.3 Stage 3: Get the result

To see the results, login the officer/ teacher account. Click “View Results”, which will redirect the page to result page.

Figure 28 Redirect to result page

Enter the private key and we will see the results.

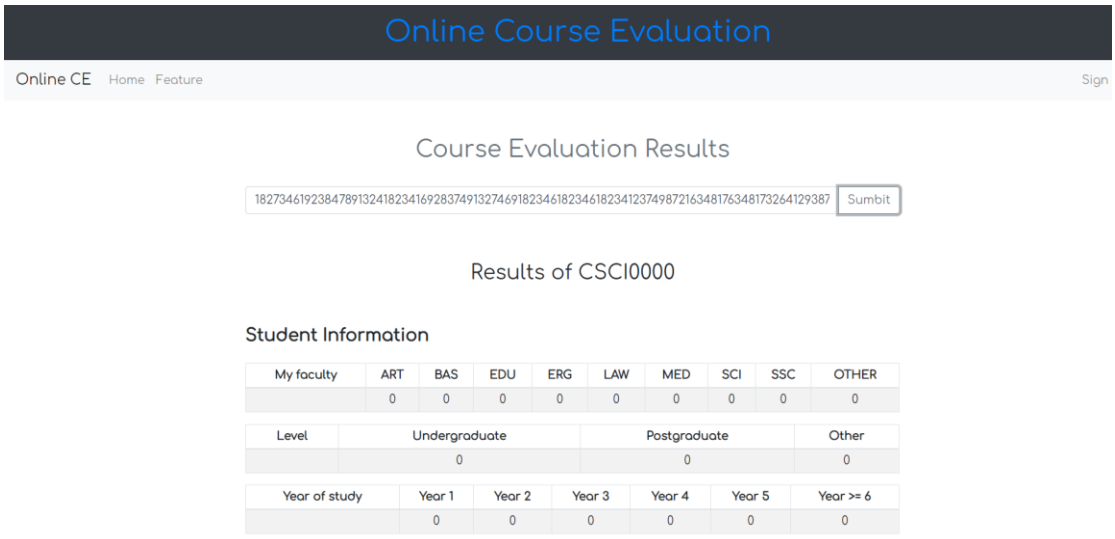


Figure 29 Results shown after entering private key

5 Individual contribution

In this project, my main contribution is the web UI design and implement the cryptosystem functions.

5.1 Web UI

For the web UI design part, I try to keep the design simple and the layout does not have complex structure. While the web UI is developed with React, I define each component according to its function. For example, the web header, web navbar, different web contents and different popups are all defined as separate components.

When I make the popup, which is used for creating new evaluation, I assume that it would be easier for the users input a xlsx file of student id, instead of entering the student ids one by one. The only problem is that I do not know official format of a student id list xlsx file, and I make another assumption for the format, which is a xlsx file with a single column of student ids. To read the data from xlsx file, I have used the react-excel-renderer [19].

For the evaluation form in our web, the questions on form is almost identical to the current paper evaluation form. The only difference is there is no open-ended question in our evaluation form. All question in the evaluation form is presented in the same page. I think it is more user-friendly for input checking.

5.2 Cryptosystem functions

5.2.1 Additive ElGamal Encryption

The encryption algorithm we used in our project is additive ElGamal encryption. In the previous section, I have mentioned Benaloh cryptosystem, Paillier cryptosystem and additive ElGamal cryptosystem. Here I will explain why I choose additive ElGamal cryptosystem as our cryptosystem.

Firstly, I do not choose Benaloh encryption because it has more constraints on generating public key if we want to get a correct decrypted value. If I generate the public key randomly, I will need more steps to check the public key before using it, which is not convenient.

Paillier encryption and additive ElGamal encryption are often used for e-voting. Paillier encryption is even easier to deploy but I choose additive ElGamal instead of Paillier encryption because Paillier encryption may have worse performance as the computations are done modulo N^2.

5.2.2 Implementation

When I implement the cryptosystem functions, BigInteger.js [20] is used for handling big integer calculation as native JavaScript library does not have good support on big integer calculation.

I have tested several versions with different sized prime number. Although bitlength of prime number is important to the security level of the encryption for defending the brute force attack, it may cost more performance. It is necessary to balance the performance and security with some testing.

5.2.2.1 Implementation with 1024 bits prime

There are seven functions to handle seven different operations:

1. Big prime number, generator, private key and public key generation
2. Encryption
3. Decryption
4. Ciphertext summation

The length of big prime is about 1024 bits. The generator and private key are randomly generated from the range from 1 to (big prime -1). The message is encrypted with a random integer generated from the 1 to 100. When we decrypt the ciphertext, brute force approach is used for solving the discrete logarithm problem. The message to be encrypted is always the number of votes to specific option, which would not be a big number and brute force approach can find it easily.

5.2.2.2 Implementation with 2048 bits prime

The length of big prime is about 2048 bits. The generator is randomly generated from the range from 1 to (big prime -1) and private key is randomly generated from the range from 1 to (big prime -1). The message is encrypted with a random integer generated from the 1 to 100.

5.2.2.3 Implementation with 512 bits prime

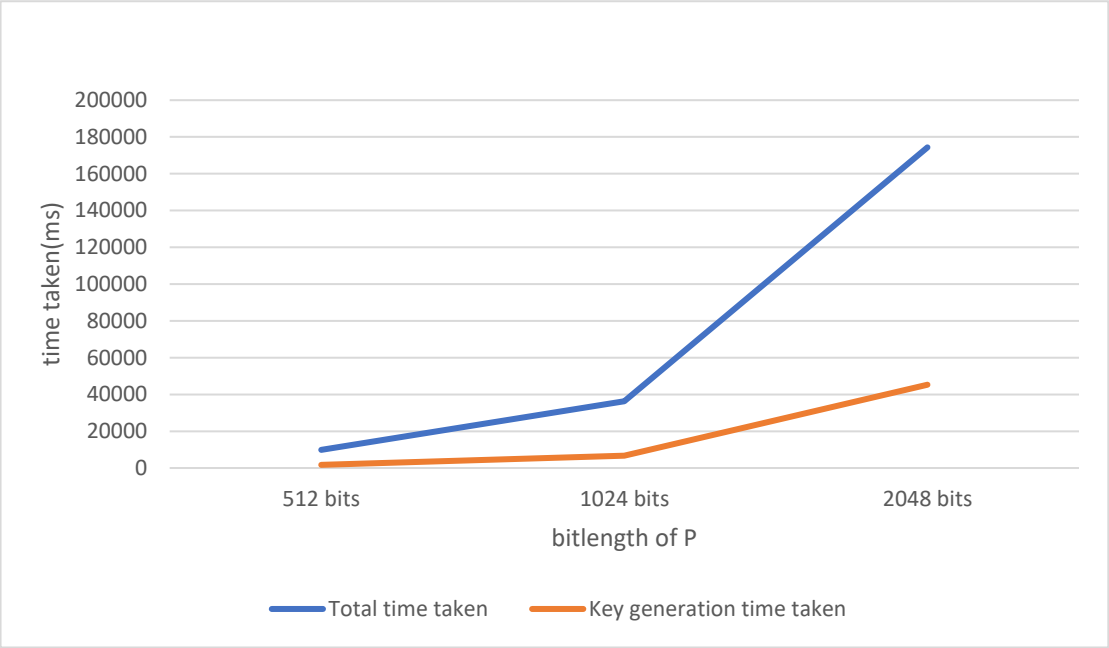
The length of big prime is about 512 bits. The generator is randomly generated from the range from 1 to (big prime -1) and private key is randomly generated from the range from 1 to (big prime -1). The message is encrypted with a random integer generated from the 1 to 100.

5.2.2.4 Performance testing

To test the performance, I run all the key generation processes and encrypt "1" with 200 times and sum all the encrypted "1". Then decrypt and output the sum. The performance will be based on the time used for finishing the whole process. This testing is run on my local computer with Chrome 81.0.4044.122.

Here are the results of testing

	Total time taken(ms)	*Keys generation time taken(ms)*
512 bits	9862	1713
1024 bits	36307	6756
2048 bits	174344	45293



Based on the results, the time taken is increased exponentially with the bitlength of prime number used. The 2048 bits prime number is secure for ElGamal encryption while the 512 bits or 1024 bits prime number may not be secure enough. In my testing, the implementation with 2048 bits prime has taken too much time to complete and it may not be practical to use it. For the one with 1024 bits prime, I could feel obvious processing time when I was generating the keys. It would not be a good user experience. My only remaining choice is the one with 512 bits prime number as it would not bring too much burden on performance.

My cryptosystem function is not optimized, and they may not fulfill all requirements for a standard additive ElGamal cryptosystem. Therefore, the performance testing may not reflect real performance of optimized additive ElGamal cryptosystem.

Using ElGamal with elliptic curve cryptography (ECC) should be a better solution for my cryptosystem. Unfortunately, I still do not understand how to implement additive ElGamal encryption with ECC and I cannot try this approach on in our project.

6 Conclusion

6.1 Term 2 Summary

In Term 1, we have set some Term 2 targets and now we have finished the targets. We have redesigned the web UI. The web UI design is not very fancy, but it should be convenient to use. The homomorphic data encryption is implemented and now we can tally the evaluation results without decryption.

Our course evaluation system now is usable, but it is not ready to be used for our course evaluation. In our report, there is no part which mentions the system testing. It is because we have not tested it with formal stress testing and security testing. The main reason is that we do not know how to stimulate a testing environment correctly. We still need some time to test out the stability of our system.

Our design may need lots of improvement before it is ready to replace the traditional paper-based course evaluation. No matter what, we will have no chance to experience the online course evaluation. We hope that someone can continue the development of anonymous course evaluation. In the future, there should be a real usable anonymous course evaluation system in CUHK.

6.2 Future Improvement

Our system still needs many improvements. Here are some possible enhancements which should be done.

6.2.1 Cryptosystem

Current implementation of cryptosystem is not performance optimized. If the additive ElGamal encryption can implemented with ECC, the security should be improved, and the encrypted data size should be smaller with smaller key size.

6.2.2 Data Schema

For our current evaluation form data schema, we stored all the options status of all questions. As a result, the submitted evaluation form data is always large when all the options value is encrypted. If we can present the data in different format, the data may become smaller. Then network can have less chance to be congested and the system will be more stable.

6.2.3 Anonymity

Although our project title is “Anonymous Online Course Evaluation”, the system is not truly anonymous. The submitted evaluation data and participants identities are not verified by zero-knowledge proof mechanism. Our system rely on token authentication and it would be a problem if an attacker can get the information of token and its owner.

References

[1] S. Zhang, J. Lee, "Analysis of the main consensus protocols of blockchain," ICT Express, 12 8 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240595951930164X [Accessed: 2 12 2019].

[2] S. Cocco, G. Singh, "Top 6 technical advantages of Hyperledger Fabric for blockchain networks," developer.ibm.com, 18 3 2018. [Online]. Available: https://developer.ibm.com/articles/top-technical-advantages-of-hyperledger-fabric-for-blockchain-networks/. [Accessed 2 12 2019].

[3] Hyperledger, "Peers," in Hyperledger Fabric Doc, [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/peers/peers.html. [Accessed 2 12 2019].

[4] "Online Course Evaluations | Registrar's Office," Stanford University, [Online]. Available: https://registrar.stanford.edu/students/online-course-evaluations. [Accessed 2 12 2019].

[5] "Helios Voting," Helios, [Online]. Available: https://heliosvoting.org/. [Accessed 2 12 2019].

[6] "Privacy," Helios, [Online]. Available: https://heliosvoting.org/privacy. [Accessed 2 12 2019].

[7] "Follow My Vote: The Online Voting Platform of The Future," Follow My Vote, [Online]. Available: https://followmyvote.com/. [Accessed 2 12 2019].

[8] "Elliptic Curve Cryptography & Online Voting," Follow My Vote, [Online]. Available: https://followmyvote.com/online-voting-technology/elliptic-curve-cryptography/. [Accessed 2 12 2019].

[9] "Polys — Online Voting System," Polys, [Online]. Available: https://polys.me/. [Accessed 2 12 2019].

[10] M. Riveiro, "How does Polys achieve anonymity and validate voters?," Polys, [Online]. Available: https://docs.polys.me/en/articles/1740042-how-does-polys-achieve-anonymity-and-validate-voters. [Accessed 2 12 2019].

[11] J. Benaloh, "microsoft.com," 2 1999. [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/1999/02/dpe.pdf. [Accessed 22 4 2020].

[12] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology — EUROCRYPT '99*, 1999.

[13] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE TRANSACTIONS ON INFORMATION THEORY,* p. 469–472, 7 1985.

[14] "Introduction," Hyperledger Fabric Doc, [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html#. [Accessed 2 12 2019].

[15] Hyperledger, "fabric-chaincode-node," GitHub repository, 2019. [Online].

Available: https://github.com/hyperledger/fabric-chaincode-node.

[16] Hyperledger, "fabric-sdk-node," GitHub repository, 2019. [Online]. Available: https://github.com/hyperledger/fabric-sdk-node.

[17] "CouchDB as the State Database," Hyperledger Fabric Doc, [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/couchdb_as_state_database.html. [Accessed 2 12 2019].

[18] Y.F. Yau; Y.K. Lui, "Final Year Project Report (Term 1) Anonymous Online Course Evaluation," 2019.

[19] ashishd751, "Quickly Render and Display Excel Spreadsheets on a Webpage with React JS".

[20] "big-integer - npm," [Online]. Available: https://www.npmjs.com/package/big-integer. [Accessed 24 4 2020].

[21] Uriel Feige, Amos Fiat, Adi Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology,* vol. 1, no. 2, p. 77, 1988.

[22] "React Bootstrap," [Online]. Available: https://react-bootstrap.github.io. [Accessed 2 12 2019].

[23] "Bootstrap," [Online]. Available: https://getbootstrap.com/. [Accessed 2 12 2019].

[24] "React – A JavaScript library for building user interfaces," [Online]. Available: https://reactjs.org/. [Accessed 2 12 2019].

[25] J. Benaloh, R. Rivest, P. Y. A. Ryan, P. Stark, V. Teague and P. Vora, "End-to-end verifiability," arXiv preprint arXiv:1504.03778, 2014.